EMERSON RIBEIRO DE MELLO

UM MODELO PARA CONFIANÇA DINÂMICA EM AMBIENTES ORIENTADOS A SERVIÇO

FLORIANÓPOLIS
2009

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PROGRAMA DE PÓS-GRADUAÇÃO

EM ENGENHARIA ELÉTRICA

UM MODELO PARA CONFIANÇA DINÂMICA EM AMBIENTES ORIENTADOS A SERVIÇO

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

EMERSON RIBEIRO DE MELLO

Florianópolis, fevereiro de 2009.

UM MODELO PARA CONFIANÇA DINÂMICA EM AMBIENTES ORIENTADOS A SERVIÇO

Emerson Ribeiro de Mello

Esta Tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica, Área de Concentração em *Sistemas de Informação*, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.

Prof. Joni da Silva Fraga, Dr.
Orientador

Prof[a]. Kátia Campos de Almeida, Dr[a].
Coordenadora do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Joni da Silva Fraga, Dr.
Presidente

Prof. Carlos Barros Montez, Dr.

Prof. Eleri Cardozo, Ph.D.

Prof. Lau Cheuk Lung, Dr.

Prof. Luciano Paschoal Gaspary, Dr.

Aos meus pais e à Karine…

AGRADECIMENTOS

Agradeço ao meu orientador, Joni da Silva Fraga, pelos questionamentos e sugestões que muito contribuíram para o desenvolvimento deste trabalho e principalmente por sua amizade. Agradeço ao meu orientador do estágio no exterior, Aad van Moorsel, por ter me recebido em seu grupo de pesquisa, por ajudar no desenvolvimento deste trabalho e também por sua amizade.

Agradeço aos demais professores do Departamento de Automação e Sistemas pelos ensinamentos e pelos laços de amizade que construí. Agradeço aos membros da banca pelos comentários e sugestões que contribuíram para a melhora deste trabalho.

Agradeço aos amigos Alysson, Cássia, Eduardo Alchieri, Eduardo Cambruzzi, Fabiano Baldo, Fábio Favarim, Fábio Rocha, Fernando, Michelle, Paulo, Rafael, Tati, Tiago, Underléa, Zézão e muitos outros com quem compartilhei boa parte da minha vivência na sala dos doutorandos do Departamento de Automação e Sistemas (DAS) e que tornaram agradáveis esses últimos anos.

Agradeço aos meus novos colegas de trabalho no IFSC, em especial aos coordenadores Cláudia e Evandro Cantú, pelo apoio imprescindível recebido na fase final da conclusão deste trabalho.

Agradeço aos meus pais, Rubens e Lacir, meus mais preciosos bens, pelo apoio e amor a mim dedicados. Agradeço também a minha família por me ajudar sempre que precisei.

Agradeço à Karine pela dedicação, apoio, confiança e amor depositados em mim durante todo o desenvolvimento deste trabalho. Agradeço também por deixar-me fazer parte de sua vida.

Por fim, agradeço ao CNPq pelo apoio financeiro o qual possibilitou o desenvolvimento deste trabalho, além de propiciar o estágio em uma universidade no exterior o qual ajudou muito meu desenvolvimento como pessoa e como pesquisador.

Resumo da Tese apresentada à UFSC como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia Elétrica.

UM MODELO PARA CONFIANÇA DINÂMICA EM AMBIENTES ORIENTADOS A SERVIÇO

Emerson Ribeiro de Mello

fevereiro/2009

Orientador: Prof. Joni da Silva Fraga, Dr.
Área de Concentração: Sistemas de Informação
Palavras-chave: Segurança, Serviços Web, Confiança
Número de Páginas: xv + 101

Resumo: Fundamentada sobre padrões abertos, a Internet possibilitou a integração de redes de computadores formadas pelos mais diversos sistemas computacionais. Os Serviços *Web* representam uma nova geração da computação distribuída e também se valem de padrões simples e poderosos permitindo que aplicações distribuídas interajam de maneira mais eficiente e sem que haja a necessidade de intervenção humana na negociação dos mecanismos subjacentes da comunicação. Várias propostas foram lançadas por órgãos padronizadores com o intuito de prover soluções para os novos desafios de segurança introduzidos pelos Serviços *Web*, contudo em algumas áreas, como o gerenciamento da confiança, ainda não existem soluções concretas. A integração de aplicações só é possível se credenciais de segurança puderem ser consideradas válidas perante todas entidades do sistema. Isto requer um modelo que permita lidar com diferentes tecnologias de segurança subjacentes além de se preocupar com o estabelecimento da confiança entre as entidades participantes. Esta tese apresenta um modelo de segurança que visa garantir a facilidade da autenticação única *Single Sign-On* (SSO) mesmo diante de diferentes tecnologias de segurança. É apresentado ainda um modelo de confiança, aliado a um sistema de reputação, o qual permite o estabelecimento dinâmico da confiança entre as entidades que compõem o sistema distribuído. O uso de um modelo de confiança baseado no conceito das redes de confiança tornam a solução escalável e a proposição de algoritmo para a localização de caminhos de confiança cobre a principal lacuna deixada pelas principais especificações voltadas para as redes de confiança. Nesta tese é apresentada também uma análise sobre os principais algoritmos de busca para redes par a par quando aplicados para a localização de caminhos de confiança. Tal análise serviu de base para a proposição de um algoritmo próprio.

Abstract of Thesis presented to UFSC as a partial fulfillment of the requirements for the degree of Doctor in Electrical Engineering.

A MODEL TO DYNAMIC TRUST IN SERVICE ORIENTED ARCHITECTURE

Emerson Ribeiro de Mello

February/2009

Advisor: Prof. Joni da Silva Fraga, Dr.
Area of Concentration: Information Systems
Key words: Security, Web Services, Trust
Number of Pages: xv + 101

Abstract: The Internet was founded over open standards that allowed the integration of computer networks composed by heterogeneous computer systems. Web Services represent a new generation of technologies for distributed computing that were built over simple and powerful standards. Organizations like W3C and OASIS released several proposals focusing on new security challenges introduced by Web Services, however there are not concrete solutions to some specific areas like trust management. Each security domain crossed by a distributed application may provide its own set of security credentials, based in its underlying security technology and policies. The diversity of underlying security mechanisms represents a challenge to integrate applications and a solution to this environment needs to make security credentials valid over all entities present in the distributed system and this solution has to provide a way that allows the trust establishment between these entities. This thesis presents a security model that aim the *Single Sign-On* (SSO) even in the presence of different security technologies. We proposed a trust model and a reputation system that allow dynamic trust establishment between entities in a distributed system. The proposed trust model is based on web of trust concept and it showed as a good solution in large scale environment. This thesis introduces an analysis over the use of P2P search algorithms to discovery trust paths and it also presents our proposal to discovery trust paths.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas

AC Autoridade Certificadora

AGD Autoridade de Gerência do Domínio

AES *Advanced Encryption Standard*

AOS Arquitetura Orientada a Serviço

B2B *Business to Business*

CORBA *Common Object Request Broker Architecture*

DCE *Distributed Computing Environment*

DCOM *Distributed Component Object Model*

DHT *Distributed Hash Table*

HTTP *HyperText Transfer Protocol*

ICP Infra-estrutura de Chave Pública

IDL *Interface Definition Language*

IdP *Identity Provider*

LDAP *Lightweight Directory Access Protocol*

OSF *Open Software Foundation*

P2P *Peer-to-Peer*

P3P *Platform for Privacy Preferences*

PDP *Policy Decision Point*

PEP *Policy Enforcement Point*

PGP *Pretty Good Privacy*

PKCS#7 *Public Key Cryptography Standard #7*

RMI *Remote Method Invocation*

RSS *Really Simple Syndication*

SAML *Security Assertion Markup Language*

SAO Serviço Agregador de Opiniões

SDSI *Simple Distributed Security Infrastructure*

SHA-1 *Secure Hash Algorithm 1*

SOA *Service Oriented Architecture*

SPKI *Simple Public Key Infrastructure*

SRO Serviço de Registro de Opiniões

SSL *Secure Sockets Layer*

SSO *Single Sign-On*

STS *Security Token Service*

TCB *Trusted Computing Base*

TCSEC *Trusted Computer System Evaluation Criteria*

TLS *Transport Layer Security*

TPC Terceira Parte Confiável

TTL *Time To Live*

UDDI *Universal Description, Discovery and Integration*

URI *Uniform Resource Identifier*

UUID *Universal Unique IDentifier*

XACML *eXtensible Access Control Markup Language*

XML *eXtensible Markup Language*

XMLDSign *XML Signature*

XMLEnc *XML Encryption*

WS *Web Services*

WSDL *Web Services Description Language*

Capítulo 1

Introdução

A Internet é conhecida por agregar os mais diversos sistemas computacionais, que variam desde a arquitetura de máquina, sistema operacional até aplicativos finais de usuários. O sucesso em um ambiente tão heterogêneo se deu através da definição de uma pilha de protocolos padronizada o que garantiria assim a interoperabilidade entre as aplicações, não importando em qual sistema operacional ou arquitetura de máquina esta estivesse rodando.

A necessidade da realização de negociações comerciais mais eficientes exigiu que um novo paradigma para o desenvolvimento de aplicações distribuídas fosse criado. A Internet surgiu diante de empresas que já faziam uso de seus sistemas computacionais e que geralmente não foram desenvolvidos para serem interoperáveis com os sistemas computacionais de seus clientes e fornecedores.

A Arquitetura Orientada a Serviço (AOS) consiste em um paradigma o qual enfatiza a descoberta dinâmica de serviços, a composição e a interoperabilidade entre esses serviços. Os Serviços *Web* apresentaram-se como a tecnologia que implementa os conceitos definidos pela AOS. Por serem transparentes para plataformas e por possuírem um modelo fracamente acoplado, os Serviços *Web* tornam-se ideais para a integração de aplicações. Sua principal vantagem foi justamente o uso de padrões abertos como o HTTP e o XML, permitindo assim que aplicações sejam integradas através de linguagens e protocolos amplamente aceitos.

Juntamente com as facilidades para a implantação dos Serviços *Web* surgiram novos desafios de segurança. A dinâmica do ambiente e a garantia da interoperabilidade são os principais requisitos para as soluções de segurança voltadas para esta área. Segundo Singhal et al. [2007] a segurança para os Serviços *Web* é dividida em quatro dimensões: troca segura de mensagens; proteção dos recursos; negociação de contratos e gerenciamento de confiança.

A troca segura de mensagens visa garantir que as mensagens trocadas entre os serviços terão garantidas as propriedades básicas de segurança, como a integridade, confidencialidade e disponibilidade. Devido ao fato do HTTP ser geralmente empregado para o transporte das mensagens, o uso do TLS/SSL [Dierks e Allen, 1999; Freier et al., 1996] torna-se uma

escolha natural. Contudo, tais protocolos não cobrem todas as necessidades introduzidas por esse novo paradigma, o que motivou a concepção de soluções específicas. Foram lançadas algumas especificações [Bartel et al., 2002; Imamura et al., 2002; OASIS, 2004b] visando a troca segura de mensagens e estas são hoje amplamente utilizadas.

A proteção dos recursos consiste em garantir que os recursos providos só estejam acessíveis às partes autorizadas. A literatura já apresentava diversos modelos e mecanismos de controle de acesso, contudo para o ambiente dos Serviços *Web* tais mecanismos não são suficientes, uma vez que, a interoperabilidade não foi considerada na concepção dos mesmos. Diante disto surgiram mecanismos de controle de acesso específicos para os Serviços *Web* [OASIS, 2005b,d] e em conjunto com os padrões para a troca segura de mensagens, conseguem cobrir parte das necessidades de segurança das aplicações voltadas para Arquitetura Orientada a Serviço (AOS).

A negociação de contratos permite obter a principal característica da AOS, que é a facilidade para a automação dos processos de negócios. Os contratos entre empresas são geralmente negociados e acordados bem antes da implementação dos sistemas. Os Serviços *Web* permitem fazer tal negociação sem a antecedência indicada, porém é necessário que exista alguma padronização para a condução de tal tarefa.

Por fim, o gerenciamento da confiança visa prover meios para permitir a integração de Serviços *Web* providos por diferentes domínios administrativos. Esta abordagem está fundamentada em modelos de confiança que fazem uso de uma Terceira Parte Confiável (TPC) ou mesmo de modelos baseados na confiança direta entre duas partes, não dependendo assim de qualquer entidade intermediária. Os modelos baseados na confiança direta entre dois pares consistem na forma mais simples, contudo menos escalável. Os modelos de confiança federados surgem como uma solução para as dificuldades apresentadas pelo modelo baseado na confiança direta. A escalabilidade em tal modelo é obtida através da delegação da confiança a uma entidade central da federação, cabendo a esta mediar a confiança com outras federações.

1.1 Motivação

A explosão da quantidade de serviços oferecidos através da Internet foi o principal ponto de motivação deste trabalho. A grande quantidade de usuários e provedores de serviços, cada qual respeitando suas próprias políticas administrativas e também de segurança, está tornando cada vez mais custoso operar neste meio.

Para os usuários o problema está em como gerenciar suas informações pessoais distribuídas por diversos provedores de serviços. A tarefa de sempre ter que fornecer o mesmo conjunto de dados para todo serviço com quem queira interagir é parte do problema. A outra parte está relacionada a privacidade de suas informações pessoais. Como garantir que abusos

sobre tais informações não serão realizados pelos provedores, como o fornecimento dessas a terceiros.

Para os provedores de serviço, uma das dificuldades consiste na grande quantidade de informações relacionada a cada um de seus usuários e que estes provedores devem gerenciar. Há também a necessidade de integrar as aplicações com outros provedores de serviço, sendo isto uma tendência do mercado para se manter competitivo. Por exemplo, o sistema de uma loja de comércio eletrônico deve considerar uma integração com a aplicação da administradora de cartões de créditos para que possa assim aumentar sua carteira de clientes.

Os Serviços *Web* surgiram com o principal objetivo de se tornar uma tecnologia integradora, visando suprir as dificuldades apresentadas por soluções anteriores para a composição de aplicações distribuídas de larga escala. Assim, os sistemas distribuídos atuais tendem a propor facilidades para o gerenciamento de identidade, tanto para os usuários quanto para os provedores de serviço. Visam usufruir das facilidades fornecidas pela arquitetura dos Serviços *Web* para garantir a integração dos sistemas, mas requerem modelos e mecanismos que garantam as necessidades de segurança de forma bilateral, dando agora uma maior importância aos desejos dos usuários, algo que geralmente não é tratado nos moldes tradicionais.

Neste ambiente, a confiança assume um papel fundamental para permitir a interação tanto entre clientes e provedores de serviços quanto somente entre provedores. Em uma aplicação distribuída a confiança pode assumir diversos sentidos, como por exemplo, garantir a identidade das partes, ou ainda para servir de base para que um cliente, diante de diversos provedores de serviço, possa escolher com qual deverá interagir. A diversidade de clientes e provedores de serviços indica um ambiente dinâmico onde as relações de confiança são facilmente estabelecidas e desfeitas. Isto sugere o uso de modelos de confiança que também permitam uma flexibilidade na gerência de tais relações, como as redes de confiança.

Patil e Shyamasundar [2005] afirma que o estabelecimento da confiança entre partes estranhas consiste em um processo complexo e subjetivo, mesmo na vida real. Em um ambiente ausente das interações humanas o desafio é ainda maior. Neste caso, a intermediação da confiança por uma parte a qual todos confiam torna-se uma saída interessante. Nas redes de confiança, essa mediação pode ser obtida através do conceito de *caminhos de confiança*, o qual interliga duas partes quaisquer, através de um caminho de confiança composto por um ou mais intermediários confiáveis. Contudo, a principal dificuldade que recai sobre tal modelo está em localizar estes caminhos.

1.2 Objetivos

O objetivo geral desta tese de doutorado consiste na proposição de um modelo de segurança voltado para o ambiente dos Serviços *Web*. Através dos padrões de segurança para os

Serviços *Web*, as propriedades básicas de segurança são garantidas nas interações entre clientes e provedores de serviço. Já através do modelo de confiança proposto nesta tese, é garantida a transposição de credenciais de segurança, possibilitando a integração de diferentes domínios administrativos e de segurança.

O modelo proposto é atingido através da combinação de alguns objetivos específicos. Inicialmente buscou-se definir as entidades que compõem a arquitetura e os relacionamentos entre essas. Isto delimitou os demais objetivos específicos, sendo estes:

- Desenvolvimento de meios para permitir a integração de aplicações que fazem uso de diferentes tecnologias de segurança, de forma que garanta a interoperabilidade entre tais tecnologias;
- Conceber um sistema de gerência da confiança, aliado a um sistema de reputação, que sirva de ferramenta de apoio a decisão às entidades da arquitetura. Tal sistema permitiria que clientes privilegiem provedores de serviços que tenham sido corretos anteriormente e que também puna os maus provedores;
- Criação de meios para permitir o estabelecimento dinâmico da confiança. O modelo de confiança nesta tese está fundamentado sobre redes de confiança, e assim sendo, o estabelecimento de novas relações de confiança entre partes estranhas está condicionado à existência de caminhos de confiança entre estas. Diante da ausência de uma solução para a localização de tais caminhos, a proposta deve ser baseada na literatura de algoritmos de busca usados em outras áreas e que possam servir às nossas intenções e permitir um estudo para a concepção de um algoritmo de busca próprio.

É objeto desta tese a implementação de um protótipo com o intuito de verificar se a arquitetura proposta é factível. Para o modelo de confiança e o sistema de reputação as nossas pretensões consistem na realização de simulações de modo a averiguar a eficiência dos mesmos. Já para os algoritmos de busca empregados na localização de caminhos de confiança, a eficiência e eficácia dos mesmos será obtida através de simulações e confrontando os resultados obtidos.

1.3 Organização do texto

Esta tese está estruturada em seis capítulos. Neste capítulo foi descrito o contexto geral do trabalho, a motivação para sua concepção bem como os objetivos desejados. Os demais capítulos encontram-se na seguinte forma:

No capítulo 2 são apresentados os principais conceitos envolvidos para o desenvolvimento deste trabalho. Buscou-se introduzir os conceitos básicos da área de segurança computacional, apresentar os conceitos por trás da Arquitetura Orientada a Serviço (AOS), bem

como os Serviços *Web* e por fim, apresentar as formas de gerenciamento de identidade e de confiança em tal ambiente.

O capítulo 3 apresenta a descrição geral do modelo proposto nesta tese. O objetivo deste capítulo é introduzir o leitor no contexto o qual este trabalho busca apresentar soluções. Neste capítulo são apresentadas as entidades que compõem o modelo e o relacionamento entre estas, além de indicar as áreas que serão cobertas pelos capítulos subseqüentes.

O capítulo 4 introduz o modelo de confiança, aliado a um sistema de reputações, utilizado para determinar os pesos associados às relações de confiança entre as entidades do modelo. São apresentados ainda os resultados obtidos através de simulações para verificar a eficiência do sistema de reputações.

O capítulo 5 mostra inicialmente as dificuldades para localizar caminhos de confiança em modelos que seguem o conceito das redes de confiança. O capítulo visa mostrar o desempenho de algoritmos de busca, empregados em redes par a par, para a localização de caminhos de confiança. É então apresentado o algoritmo *DiffTrust*, proposto nesta tese, além dos resultados obtidos através de simulações.

O capítulo 6 apresenta as conclusões colhidas com esta tese, além de ilustrar as perspectivas futuras para a continuação deste trabalho.

Capítulo 2

Conceitos gerais

Neste capítulo serão introduzidos os conceitos necessários para a compreensão dos demais capítulos desta tese. A seção 2.1 introduz os conceitos básicos de segurança, além de apresentar os conceitos de segurança em sistemas distribuídos, que é o principal foco desta tese. A seção 2.2 apresenta os principais componentes da arquitetura orientada a serviços. Por fim, na seção 2.3 são apresentadas as formas para o gerenciamento de identidades e de confiança no ambiente dos Serviços *Web*.

2.1 Segurança em Sistemas Distribuídos

2.1.1 Princípios básicos de segurança computacional

A segurança é vista como uma qualidade de serviço que garante o provimento de recursos mesmo diante de ações de indivíduos não autorizados. A segurança está fundamentada sobre três propriedades que devem ser mantidas [Russell e Gangeni, 1991]:

- Confidencialidade - A informação só deve ser revelada para usuários autorizados a acessá-la;
- Integridade - A informação não poderá ser modificada, intencionalmente ou acidentalmente, por usuários que não possuam direito para tal;
- Disponibilidade - O uso do sistema não poderá ser negado, de forma maliciosa, a usuários autorizados;

Em alguns trabalhos na literatura, como em Landwehr [2001], também citam o ***Não-repúdio*** como uma propriedade de segurança. O Não-repúdio assegura que um usuário não poderá negar a sua participação na ocorrência de um evento ou transação.

2.1.2 Autenticação e Autorização

O processo de identificação em sistemas computacionais consiste de um conjunto de procedimentos e mecanismos que permitem que agentes externos (usuários, dispositivos, etc) sejam identificados como principais autorizados segundo as políticas de segurança adotadas no sistema. No processo de identificação, alguns mecanismos de segurança exigem um nome de usuário e uma senha, assim garantem a identidade do agente externo, dando a este a possibilidade de usufruir do sistema. Já o processo de autenticação, composto por um conjunto de mecanismos e procedimentos, dá ao sistema a possibilidade de assegurar que um principal[1] é realmente quem ele diz ser.

Se considerado um sistema distribuído, em uma comunicação entre duas partes, o serviço de autenticação preocupa-se em assegurar autenticidade da comunicação. Neste caso, dois aspectos são envolvidos. Primeiro, no início da conexão, o serviço de autenticação garante a autenticidade das duas entidades (autenticação mútua). Segundo, o serviço deve assegurar que a comunicação não é interferida de tal maneira que uma terceira parte não autêntica consiga personificar uma das duas partes autênticas com o propósito de transmitir ou receber informações de forma não autorizada. Logo, um serviço de autenticação pode promover a identificação de principais, a autenticação mútua entre as partes e a ainda garantir a autenticidade dos dados transmitidos entre as partes.

A autorização é o processo que envolve obter algum tipo de acesso sobre um recurso. No caso de um sistema multiusuário, o administrador do sistema define quais usuários terão permissão para acessar algum recurso do sistema (arquivos, impressoras, etc) e assim sendo somente os usuários com os devidos direitos poderão acessar os recursos protegidos. Os mecanismos de controle de acesso são concretizados normalmente por meio de modelos que usufruem de um *monitor de referência* (ver figura 2.1). O objetivo do monitor de referência é intermediar toda tentativa de acesso feita por usuários (ou programas representado-os) a recursos do sistema. Uma base de dados contém todas as autorizações (podendo ser na forma de uma matriz de acesso) as quais são atribuídas de acordo com a política de autorização do sistema.

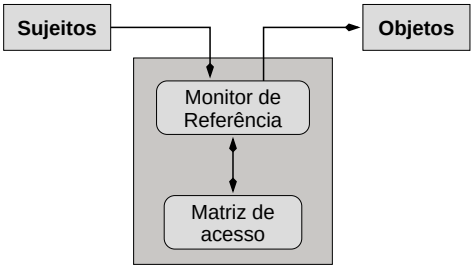


Figura 2.1: Monitor de referência

[1] Termo usado para identificar usuários, processos ou máquinas atuando em nome dos usuários de um sistema, que são considerados aptos pela política de segurança.

A implementação destes mecanismos de autenticação e autorização em sistemas distribuídos é uma tarefa complexa. Serviços como autenticação e autorização, serviço de nomes, comunicação, entre outros, são influenciados pela escalabilidade. Em ambientes de larga escala, até mesmo a habilidade do usuário interagir com um serviço é afetada.

Um sistema é dito escalável se o mesmo puder tratar a adição de usuários e recursos sem sofrer uma perda notável de desempenho ou um aumento na complexidade de administração [Neuman, 1994]. Na literatura [Housley et al., 2002; Ellison et al., 1999; DoD, 1985] são apresentadas e discutidas algumas abordagens para a implementação dos mecanismos de autenticação e autorização: abordagem centralizada, autenticação centralizada e autorização descentralizada, abordagem descentralizada.

Abordagem Centralizada

A Base Computacional Confiável (*Trusted Computing Base* – TCB) [DoD, 1985] consiste em uma pequena quantidade de *software* e *hardware*, da qual depende a segurança do sistema [Lampson et al., 1992] e está presente em uma única máquina (serviços de autenticação e autorização centralizados). A vantagem trazida com tal abordagem é a facilidade na administração do sistema. Porém, em sistemas distribuídos a centralização desses serviços acarreta em uma perda de desempenho, pois uma única máquina seria responsável por autenticar e autorizar todos os pedidos originados no sistema, tornando-a um gargalo nas comunicações. A centralização dos serviços em uma única máquina cria um ponto único de vulnerabilidade ou de falha, pois se a máquina vier a falhar todo o sistema fica comprometido. Há ainda a necessidade de que todas as máquinas do domínio tenham uma relação de confiança com o servidor de autenticação e autorização. Tais características tornam esta abordagem não muito atrativa para sistemas distribuídos de larga escala.

Autenticação Centralizada e Autorização Descentralizada

Nesta abordagem, os controles de autenticação continuam centralizados em uma única máquina do sistema e a autorização é realizada sobre os objetos (locais) de cada máquina do sistema. Tal abordagem torna o sistema menos dependente de uma única máquina, porém o ponto único de vulnerabilidade ainda está presente (serviço global de autenticação). É necessário ainda que haja uma relação de confiança entre o servidor de autenticação e as demais máquinas. O fato de possuir o controle descentralizado traz dificuldades em manter a coerência da política de autorização.

A escalabilidade é alcançada por esta abordagem através da introdução da noção de *domínios de segurança*. Esses domínios são caracterizados por um servidor de autenticação centralizando as políticas de autenticação. Servidores de autenticação podem possuir relações de confiança entre si, permitindo a troca entre domínios das certificações de autenticação e

dos controles de autorização, o que permite ao modelo atuar em ambientes de larga escala. O X.509 [Housley et al., 2002] baseia-se neste tipo de abordagem.

Abordagem descentralizada

A autenticação e autorização são realizadas por cada máquina do domínio, não havendo mais um ponto único de vulnerabilidade. Entretanto ainda há dificuldade em manter a coerência das políticas de autorização. Uma das formas de se implementar essa abordagem é o uso das redes de confiança. O *Simple Public Key Infrastructure* (SPKI)/*Simple Distributed Security Infrastructure* (SDSI) [Ellison et al., 1999; Rivest e Lampson, 1996] e o *Trusted Computer System Evaluation Criteria* (TCSEC) [DoD, 1985] adotam esta abordagem. Nas redes de confiança, os sujeitos, considerados confiáveis, implantam as políticas de autenticação e autorização, e encontram-se distribuídos pela rede.

2.2 Arquitetura Orientada a Serviços

A Arquitetura Orientada a Serviço (AOS) (*Service Oriented Architecture* – SOA) consiste em uma caracterização de sistemas distribuídos, que visa reorganizar aplicações e sua infraestrutura, através de um conjunto de interações de serviços que são acessados através de interfaces e protocolos padronizados, tendo como foco processos de negócio [Papazoglou, 2003].

A AOS define três tipos de papéis: *Diretório para registro de serviços* – repositório que é utilizado para publicação e localização de interfaces dos serviços; *Provedor de serviços* – entidade responsável por publicar as interfaces dos serviços, providos por este, no registro de serviços e também responsável por atender as requisições originadas pelos clientes; e *Cliente* – aplicação ou um outro serviço que emite requisições a um serviço. Cada participante da arquitetura pode ainda assumir um ou mais papéis, podendo ser por exemplo, um provedor e um cliente de serviços.

A AOS também descreve o relacionamento entre estes papéis, especificando as operações: *publicar*, *localizar* e *invocar*. A figura 2.2 ilustra a colaboração entre os participantes da AOS, onde o cliente efetua uma busca por um serviço, especificando as características desejadas, ao diretório de registros. Se o serviço existir, então é retornado para o cliente a interface e a localização do respectivo serviço. Por fim, o cliente faz uma invocação ao provedor do serviço.

Os serviços estão baseados nas trocas de mensagens entre os provedores e clientes. As mensagens seguem um formato padrão garantindo aos serviços a neutralidade da tecnologia, permitindo que provedores e clientes utilizem diferentes implementações nas camadas inferiores.

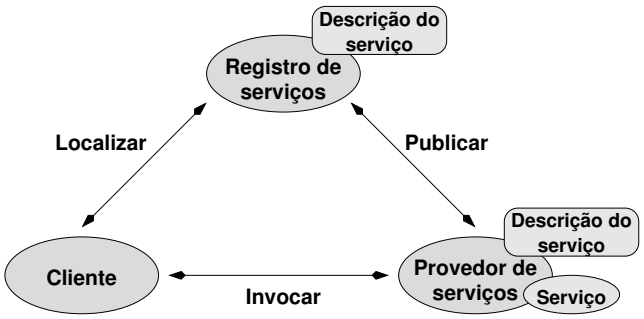


Figura 2.2: Interação entre entidades da AOS

As interfaces dos serviços são auto-descritivas e baseadas em padrões abertos. A interface de um serviço define um conjunto de métodos públicos, juntamente com seus parâmetros, valores de retorno e meios para tratar possíveis exceções, porém não provê uma implementação. A interface é um contrato entre o provedor do serviço e o cliente, sendo que o provedor deverá implementar todos os métodos ali descritos, e o cliente só poderá invocar tais métodos.

Por estarem relacionados diretamente às funções de negócios, os serviços representam uma forma de modularidade diferente daquelas existentes nas linguagens de programação como os módulos, componentes e objetos. Componentes representam entidades e regras de negócio, um serviço representa uma função de negócio completa, sendo composto por uma coleção de componentes. Serviços podem ser reutilizados e empregados em novas transações, na camada de negócios, dentro de uma organização ou através de organizações.

2.2.1 Arquitetura dos Serviços *Web*

Os Serviços *Web* (*Web Services* – WS) são classificados como um tipo específico de serviço, o qual é identificado através de um identificador uniforme de recursos (*Uniform Resource Identifier* – URI). São independentes de linguagens de programação, sistemas operacionais e das arquiteturas de máquinas. Sua principal característica é a utilização de padrões abertos como o XML e o HTTP. Através do uso de padrões abertos, os Serviços *Web* conseguem garantir a interoperabilidade entre clientes e provedores de serviços, sem que os mesmos necessitem possuir o conhecimento prévio de quais tecnologias estão presentes em cada lado. Tal facilidade é ideal para que as relações de negócios entre empresas (*Business to Business* – B2B) sejam estabelecidas de maneira simples e dinâmica.

Serviços *Web* não constituem uma outra tecnologia baseada nos conceitos dos objetos distribuídos, como o CORBA[2] [OMG, 2002], DCOM[3] [Brown e Kindel, 1996] e RMI[4] [Sun, 2002]. Em Vogels [2003] é apresentada uma discussão sobre as semelhanças e diferenças entre Serviços *Web* e sistemas de objetos distribuídos. Para Vogels [2003], os Serviços *Web*

[2] *Common Object Request Broker Architecture*
[3] *Distributed Component Object Model*
[4] *Remote Method Invocation*

são um tipo de tecnologia de sistemas distribuídos e a única relação possível com os objetos distribuídos, é que os Serviços *Web* vem sendo algumas vezes utilizados em áreas onde as aplicações de objetos distribuídos falharam no passado.

A tecnologia de objetos distribuídos e de Serviços *Web* possuem algumas características em comum, como uma linguagem para descrição de interfaces (*Interface Definition Language* – IDL), garantindo interações de rede bem definidas; e mecanismos semelhantes para registro e localização de objetos ou serviços.

Nos sistemas de objetos distribuídos, um objeto é identificado através de uma referência, que o torna único no sistema. A noção de *referência de objetos* é essencial dentro de um sistema de objetos distribuídos, visto que objetos geralmente possuem referências para outros objetos, possibilitando assim a computação distribuída com estado. A principal diferenciação entre Serviços *Web* e objetos distribuídos é o ciclo de vida dos objetos. Um ciclo de vida de um objeto é composto pelas seguintes fases:

- Através de um pedido, uma fábrica cria uma instância de um objeto;
- O cliente que requisitou o pedido, pode agora executar operações no objeto instanciado;
- E por fim, em algum momento posterior, o cliente remove a instância do objeto que não será mais utilizado.

Os Serviços *Web* não possuem um ciclo de vida com características como, objetos, referências e fábricas. Serviços *Web* não conseguem oferecer qualquer *facilidade de estado* para computação distribuída, característica básica de um sistema de objetos distribuídos. A arquitetura dos Serviços *Web* também não define relações entre as invocações realizadas em um mesmo serviço ou ainda em serviços relacionados.

Ambientes como uma rede local, são caracterizados pela homogeneidade de plataforma e por possuírem um tempo máximo de latência conhecido. Tal tipo de ambiente é ideal para a tecnologia de objetos distribuídos, visto que é uma tecnologia madura e, dentro de tal ambiente, bem robusta. Em ambientes como a Internet, onde a interoperabilidade e suporte para plataformas e redes heterogêneas são essenciais, os Serviços *Web* demonstram ser os mais adequados. Porém, há a necessidade de incorporar características básicas presentes nas tecnologia de objetos distribuídos, como garantia de ordem total e a prevenção contra mensagens duplicadas.

A adoção dos Serviços *Web* não implica no uso de qualquer aplicativo adicional no cliente ou no servidor. Para o cliente, basta uma linguagem de programação que dê suporte para XML e HTTP, por exemplo. Já para o servidor, basta que o mesmo possua um servidor de aplicação para disponibilizar os serviços. Tal característica define os Serviços *Web* como *auto-contidos*. Serviços *Web* também são definidos como *auto-descritivos*, sendo que tanto

o cliente como o servidor só precisam se preocupar com o formato e com o conteúdo das mensagens a serem trocadas, abstraindo os detalhes de implementação. Um Serviço *Web* é composto basicamente por quatro elementos [Vogels, 2003]:

- **Serviço** – Um aplicativo hábil para processar documentos XML recebidos através de uma combinação de protocolos de transporte e de aplicação. Detalhes de como esse componente é construído, como técnicas de orientação a objetos, etc., não são especificados. O único requisito necessário para este tipo de componente, é que o mesmo esteja apto a tratar documentos XML;
- **Endereço** – Combinação entre protocolo e endereço de rede, utilizado para que um cliente possa acessar um serviço;
- **Documento XML** – Um documento que contém informações específicas à aplicação;
- **Envelope** – Encapsulamento que garante que documentos XML sejam processados de forma correta, separando as informações relacionadas a comunicação, dos dados em si. Por exemplo, informações relacionadas a forma como a mensagem será cifrada ou assinada podem ser especificadas em um envelope sem que o documento XML original seja modificado.

O processo para tornar um Serviço *Web* disponível publicamente, requer inicialmente que o provedor de serviços descreva a interface do serviço que deseja prover, no caso utilizando uma linguagem padrão, o *Web Services Description Language* (WSDL) [W3C, 2001] e depois publique a interface em um serviço de busca público. O *Universal Description, Discovery and Integration* (UDDI) [OASIS, 2004a] é um serviço padrão, para publicação e localização, utilizado na arquitetura dos Serviços *Web*. A comunicação entre o provedor e o consumidor de um serviço é realizada através de trocas de mensagens XML, encapsuladas dentro de envelopes SOAP [W3C, 2003].

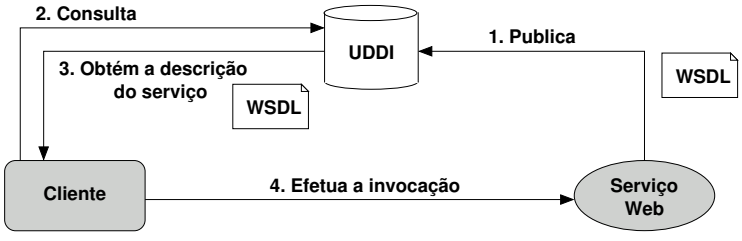


Figura 2.3: Colaboração típica na Arquitetura dos Serviços *Web*

A figura 2.3 ilustra uma colaboração típica dos Serviços *Web*, a qual está baseada no modelo da AOS apresentado na seção 2.2. No passo 1 o provedor publica a interface WSDL no serviço UDDI, tornando assim o serviço visível para os possíveis clientes. No passo 2, um cliente realiza a busca por serviços que correspondam com as necessidades informadas

e assim, no passo 3, recebe a interface WSDL do serviço que possui as características desejadas. Por fim, no passo 4 o cliente invoca o serviço desejado, respeitando a interface obtida anteriormente, sendo tal invocação através de mensagens SOAP.

2.2.2 Segurança na arquitetura dos Serviços *Web*

Os Serviços *Web* representam uma evolução de sistemas distribuídos. Permitem a integração de aplicações existentes e por geralmente utilizarem do HTTP como protocolo de transporte, conseguem ultrapassar os limites impostos pelos filtros de pacotes tradicionais (*firewalls*), limite os quais eram proibitivos para as aplicações distribuídas como, por exemplo, em CORBA [OMG, 2002]. Juntamente com as facilidades de integração, surgiram novos desafios de segurança que tornam insuficientes os atuais mecanismos de segurança. Como garantir a segurança diante de um roteamento entre múltiplos Serviços *Web* e ainda, como abstrair as tecnologias de segurança usadas nas camadas inferiores e garantir a interoperabilidade proposta pela arquitetura dos Serviços *Web*.

Os Serviços *Web* permitem que aplicações se comuniquem sem a necessidade de qualquer tipo de interação com o usuário final. Por exemplo, o sistema de uma empresa aérea poderia fornecer um serviço para seus clientes, de forma que os mesmos pudessem comprar seus bilhetes e teriam reservas em hotéis associados. Neste caso, o usuário final estaria interagindo com o sistema da empresa aérea e este, por sua vez, estaria mediando a comunicação entre o usuário e o sistema do hotel. O problema aqui está em como garantir que as informações do usuário cheguem até o sistema do hotel de forma segura, visto que, as informações sensíveis do usuário, que só interessam ao sistema do hotel, estariam sendo roteadas, e disponíveis, pelo sistema da empresa aérea.

O roteamento entre múltiplos Serviços *Web* é comumente utilizado para obter escalabilidade e também para agir como uma ponte entre diferentes protocolos de transporte. Tecnologias como o TLS/SSL [Dierks e Allen, 1999; Freier et al., 1996] permitem garantir a confidencialidade entre duas partes (na camada de transporte), porém não proporciona segurança fim a fim, uma vez que a mensagem, para atingir o destinatário final, pode passar por diversos nós intermediários, situados na camada de aplicação. Se a cifragem for empregada somente na camada de transporte, nós intermediários terão reveladas as informações que passam por estes, de forma proposital ou através das lacunas existentes (na camda da aplicação) entre uma sessão segura (na camada de transporte) e outra.

As lacunas de segurança não ocorrem no transporte dos dados, mas sim quando os mesmos estão disponíveis nos nós intermediários. Dessa forma, informações confidenciais presentes nas mensagens SOAP, que deveriam permanecer confidenciais durante todo o percurso através dos nós SOAP intermediários, poderiam ficar expostas. Para tratar tal desafio, princípios de segurança devem ser aplicados em um contexto de segurança o qual não só se restringe a uma simples troca de mensagens entre dois nós SOAP, como ilustrado

pelo 1º contexto da figura 2.4 (segurança somente na camada de transporte). O contexto de segurança deve ser mais amplo e atuar também na camada da aplicação para garantir a segurança fim a fim, como ilustrado pelo 2ºcontexto.

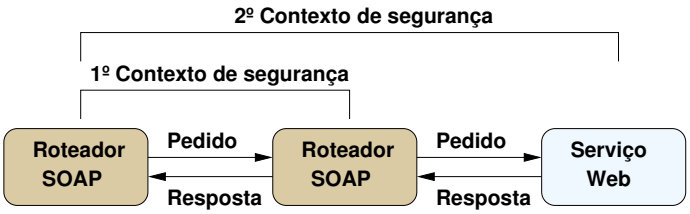


Figura 2.4: Contextos de segurança [IBM e Microsoft, 2002]

Um outro desafio é como garantir os limites de segurança, antes determinados pelos *firewalls*. Os filtros de pacotes tradicionais se preocupam basicamente com a segurança na camada de rede, analisando se o pacote vem de uma origem confiável, porém não se preocupam com o conteúdo dos pacotes. Assim, toda e qualquer requisição a um Serviço *Web* irá transpor o *firewall*. Os Serviços *Web* também estão suscetíveis a tipos de ataques já conhecidos como a negação de serviço, mensagens antigas, estouro de pilha, entre outros. Para garantir a segurança neste novo tipo de ambiente, novos mecanismos de segurança devem ser implantados também nas camadas superiores da pilha TCP/IP e devem operar em conjunto com os mecanismos presentes nas camadas inferiores.

Existem diversos requisitos de segurança para que os Serviços *Web* possam ser adotados por completo, sendo estes [W3C, 2004]:

- Mecanismos de autenticação e de autorização
- Confidencialidade e integridade fim a fim
- Integridade das transações e comunicações
- Não-repúdio
- Trilhas de auditoria
- Aplicação das políticas de segurança de forma distribuída

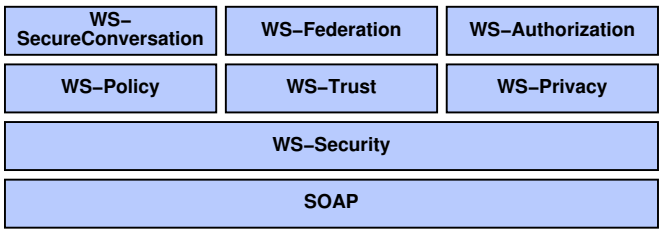


Figura 2.5: Especificações de segurança para os Serviços *Web* [IBM e Microsoft, 2002]

Diversas propostas e especificações de segurança para os Serviços *Web* foram lançadas por órgãos padronizadores, como W3C e OASIS. As propostas estão direcionadas para diversas áreas de segurança e fazem uso de algumas das especificações de segurança para XML. A figura 2.5 ilustra as principais especificações, algumas aprovadas pelos órgãos padronizadores, como a WS-Security, e outras em fase de acabamento, como a WS-Trust. Abaixo é apresentado uma pequena discussão sobre algumas especificações ilustradas pela figura 2.5.

Especificações de segurança para XML

O uso de assinaturas de digitais é uma forma para garantir as propriedades de integridade e autenticidade de informações digitais. A especificação *XML Signature* (XMLDSign) [Bartel et al., 2002], proposta conjunta dos órgãos W3C e IETF, define regras para gerar e validar assinaturas digitais expressas em XML. A XMLDSign possui pontos em comuns com a *Public Key Cryptography Standard #7* (PKCS#7), porém apresenta formas para tratar os novos desafios em se trabalhar com documentos XML.

O uso da XMLDSign não está unicamente voltado para assinar documentos XML. É possível assinar qualquer tipo de documento eletrônico (arquivos binários ou textos), sendo que a assinatura será representada através de um documento XML. Também é possível assinar somente algumas partes de um documento XML, permitindo assim que outras partes de um documento XML sofram modificações, sem que isso invalide a parte assinada. A XMLDSign não define novos algoritmos ou funções de resumos criptográficos, como o RSA [RSA, 2002] e SHA-1 [Eastlake e Jones, 2001], mas define como usar os algoritmos existentes em documentos XML.

A especificação *XML Encryption* (XMLEnc) [Imamura et al., 2002] visa prover segurança fim a fim para aplicações que necessitem realizar troca de dados de forma segura. Diferentemente dos protocolos TLS/SSL [Dierks e Allen, 1999; Freier et al., 1996], que só garantem a confidencialidade dos dados durante a sessão estabelecida entre duas partes (camada de transporte), a XMLEnc provê confidencialidade persistente, garantindo assim a confidencialidade dos dados mesmo depois do término da interação.

A XMLEnc provê soluções para algumas necessidades não cobertas pelo TLS/SSL, como a possibilidade de cifrar somente partes de um dado e o estabelecimento de comunicação segura entre mais de duas partes. Os dados cifrados são representados de uma forma estruturada e permite que em um mesmo documento esteja presente informações cifradas e não cifradas. Tal estrutura ainda possibilita o uso de diferentes chaves para cifrar partes de um documento, permitindo assim que um mesmo documento seja trocado entre diversas partes, sem que ocorra a revelação de informação para partes não autorizadas e garantindo o acesso a informação, por partes autorizadas.

De forma análoga ao XMLDsig, o XMLEnc representa, de forma estruturada, dados cifrados e permite cifrar documentos XML ou não. A estrutura do XMLEnc, além de expressar

os dados cifrados, também expressa detalhes sobre o tipo do documento cifrado (jpeg, xml, etc.); a chave simétrica que será utilizada na sessão; informações sobre o tipo da chave simétrica; e o método de cifragem utilizado (ex: RSA para cifrar a chave secreta e AES [Daemen e Rijmen, 2002] para cifrar os dados).

XACML

A autorização é uma propriedade básica de segurança que determina se um principal pode ou não executar alguma ação sobre algum recurso. Geralmente, cada sistema utiliza uma linguagem própria para definição das políticas, tornando assim um fator limitante para a concepção de sistemas distribuídos e abertos. Visando garantir a interoperabilidade entre os diversos sistemas, a OASIS lançou a *eXtensible Access Control Markup Language* (XACML) [OASIS, 2005b], um sistema de políticas de propósito geral, baseado em XML.

A XACML descreve uma linguagem para políticas de controle de acesso e também um formato para mensagens de *pedido* e *resposta*. A linguagem para política de controle de acesso é utilizada para definir quem possui direitos de acesso sobre o que. O formato de *pedido* e *resposta* descreve como as consultas sobre o sistema de políticas deverão ser realizadas (pedido) e como deverão ser as respostas.

O formato de *pedido* e *resposta* define as trocas ente o *Policy Decision Point* (PDP) [Yavatkar et al., 2000], ponto o qual efetua o processamento da política, e o *Policy Enforcement Point* (PEP) [Yavatkar et al., 2000], ponto o qual concretiza as decisões de política. A XACML foi desenvolvida para garantir a interoperabilidade entre diversas aplicações. Assim, uma camada de abstração entre o ambiente da aplicação e a linguagem núcleo do XACML é feita através de um Contexto XACML. Um Contexto XACML é definido através de um esquema XML, que descreve uma representação canônica das entradas e saídas do PDP [OASIS, 2005b].

Um pedido é composto por atributos associados ao sujeito que está originando a requisição, identificação do recurso desejado, pelas ações que serão executadas no recurso, e também pelos atributos do ambiente. Já na resposta são contidas decisões como: *permit* – para acesso garantido; *deny* – para acesso negado; *not applicable* – para a inexistência de política ou de regras associadas ao recurso; ou ainda *indeterminate* – para a ocorrência de erros durante o processamento [Lorch et al., 2003].

A figura 2.6 ilustra o fluxo de dados entre um cliente tentando acessar um recurso, utilizando-se do XACML. No passo 1 o sujeito (cliente) lança um pedido ao provedor do recurso. Este pedido é interceptado pelo PEP, o qual monta um pedido XACML e encaminha ao *tratador de contexto* (passo 2). O *tratador de contexto* encaminha o pedido ao PDP para que o mesmo decida sobre a tentativa de acesso (passo 3). O PDP pode requisitar ao *tratador de contexto*, atributos relacionados ao recurso e ao sujeito (passos 4, 5 e 6). Em posse

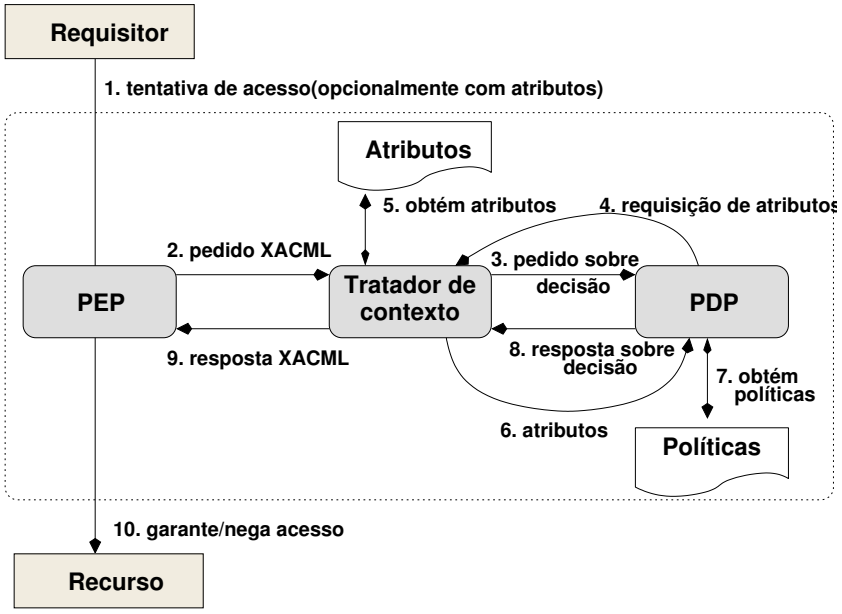


Figura 2.6: Fluxo de dados com o XACML [OASIS, 2005b]

dos atributos, o PDP requisita as políticas associadas com as entidades envolvidas (passo 7) e assim gera uma resposta sobre a decisão tomada (passo 8). O *tratador de contexto* gera uma resposta XACML e envia ao PEP (passo 9). E por fim, o PEP garante ou não o acesso ao recurso (passo 10).

SAML

A *Security Assertion Markup Language* (SAML) [OASIS, 2005d] consiste de um conjunto de especificações e esquemas XML, que juntos definem uma forma padrão para criar, trocar e interpretar asserções de segurança entre entidades de uma aplicação distribuída. A SAML define meios para expressar, em XML, informações sobre autenticação, autorização e atributos de um sujeito, porém não define uma nova tecnologia ou forma para autenticação.

A SAML está baseada no princípio da "confiança portável", o que permite a um usuário autenticar-se uma única vez (*Single Sign-On* – SSO) em alguma entidade e usufruir dos direitos concedidos com esta ação em todas as demais entidades participantes. A SAML provê um fraco acoplamento entre diretórios, sendo que as informações dos usuários não necessitam estarem dispostas e sincronizadas entre todos os diretórios das entidades participantes.

A SAML é neutra à plataforma e às tecnologias de segurança e visa garantir a interoperabilidade entre os diferentes sistemas de autenticação e de autorização. Foi proposta inicial visava trabalhar com protocolos amplamente aceitos, como o HTTP, SMTP e SOAP. Os principais componentes da SAML são:

- **Asserções** (*Assertions*) – Asserção é um conjunto de afirmações, dado por um emissor

SAML, sobre determinadas informações de um principal. A SAML define três tipos de asserções:

  - **Asserção de autenticação** – É fornecida pelo emissor SAML, após a autenticação com sucesso do usuário. Na asserção estão contidas informações como: emissor da asserção, o usuário autenticado, período de validade, etc;
  - **Asserção de atributo** – Detalhes específicos sobre o usuário, por exemplo, número do CPF;
  - **Asserção de autorização** – A posse dessa asserção indica que o usuário detentor possui as autorizações expressas por ela. Essa asserção é formada com base nas asserções de autenticação e de atributo.

- **Protocolos** – Detalha como serão realizados os pedidos e respostas sobre as asserções SAML;
- **Mapeamento** (*Bindings*) – Detalha como os protocolos SAML são mapeados nos protocolos de transporte. A especificação SAML define uma forma de mapeamento para permitir que mensagens de *pedido* e *resposta* SAML sejam transportadas em mensagens SOAP;
- **Perfis** – Define as restrições e extensões necessárias para um determinado tipo de aplicação. São definidos alguns perfis, como o *Web Browser SSO* – define mecanismos para possibilitar a autenticação única em navegadores *Web*; *Single Logout Profile* – define como o SOAP e comandos *HTTP Redirect*, *HTTP Post* podem ser usados para encerrar uma sessão; entre outros.

A figura 2.7 ilustra um exemplo de uma asserção SAML de autenticação. A asserção tem como sujeito o usuário "João", identificado pelo seu endereço eletrônico, que é um dos formatos pré-definido na especificação SAML para identificar sujeitos.

A autenticação se fez através de usuário e senha sobre uma sessão protegida (linhas 19–21) e esta ocorreu em `2005-06-22T09:11:00Z` (linha 3). A especificação da SAML prevê o uso de diferentes mecanismos para a autenticação: usuário e senha, Kerberos [Kohl e Neuman, 1993], *Secure Remote Password* [Wu, 1998], certificados TLS/SSL, chave pública (X.509 [Housley et al., 2002], SPKI [Ellison et al., 1999], XKMS [Hallam-Baker e Mysore, 2005]), XMLDSign e ainda, possibilita o uso de mecanismos não definidos na especificação.

A especificação da SAML não define as entidades responsáveis pela emissões das asserções, porém tais entidades são previstas pelo modelo de uso do SAML. A especificação define diferentes variantes do protocolo SAML de *pedido* e *resposta*, relacionados às asserções de autenticação, atributos e autorização.

Os pedidos SAML para uma autoridade de autenticação poderiam questionar se um determinado sujeito está autenticado e como resposta obter uma asserção SAML de autenticação.

```
<saml:Assertion xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   Version="2.0"
   IssueInstant="2005-06-22T09:11:00Z">
   <saml:Issuer>
      www.emissor-saml.br
   </saml:Issuer>
   <saml:Subject>
      <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
          ">
         joao@empresa.br
      </saml:NameID>
   </saml:Subject>
   <saml:Conditions
      NotBefore="2005-06-22T09:11:00ZA"
      NotOnOrAfter="2005-06-22T12:11:00Z">
   </saml:Conditions>
   <saml:AuthnStatement
      AuthnInstant="2005-06-22T09:11:00Z" SessionIndex="9876229339">
      <saml:AuthnContext>
         <saml:AuthContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
         </saml:AuthContextClassRef>
      </saml:AuthnContext>
   </saml:AuthnStatement>
</saml:Assertion>
```

Figura 2.7: Asserção SAML de Autenticação

Já para uma autoridade de atributos poderia ser questionado quais são os atributos pertencentes a um determinado sujeito autenticado, recebendo assim uma asserção SAML de atributos. Por fim, um pedido SAML de autorização poderia questionar se um determinado sujeito autenticado, e munido de atributos, poderia executar uma ação específica sobre um determinado recurso, obtendo como resposta uma asserção SAML de autorização.

Em sua primeira versão, SAML 1.0, o principal objetivo era permitir a transferência de autenticação e autorização entre aplicações *web*. Já a versão 1.1 foi lançada com o intuito de melhorar a interoperabilidade e garantir uma melhor integração com o XMLDSign. Com base nas iniciativas do projetos Liberty Alliance[5] e Internet2 Shibboleth[6], a versão 2.0 da SAML tem como foco principal o uso de identidades federadas e apresentando ainda as seguintes características [OASIS, 2005c]:

- **Pseudônimos** – Pseudônimos, ou identificadores opacos, permitem que principais interajam com o sistema sem a necessidade de revelar qualquer informação que o identifique, como e-mail, nome, etc. O uso de pseudônimos impede que provedores entrem em comum acordo para cruzar informações de um determinado principal e assim ferir sua privacidade;

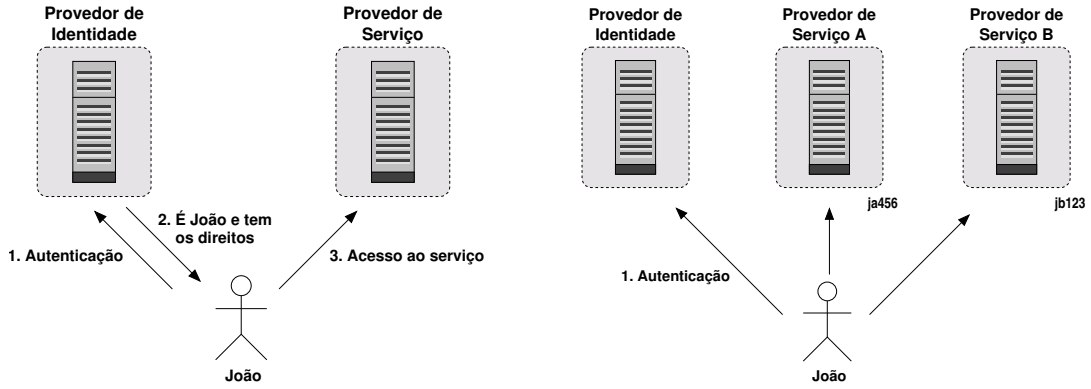
[5] http://www.projectliberty.org
[6] http://shibboleth.internet2.edu

- **Gerenciamento de identificadores** – Define como dois provedores poderão estabelecer e subseqüentemente gerenciar os pseudônimos dos principais, com quem operam;
- **Metadados** – Define como expressar dados de configuração e dados de confiança, para tornar mais simples o uso do SAML, haja visto que as entidades participantes devem aceitar os mesmos papéis, identificadores, perfis, URL e certificados;
- **Cifragem** – Possibilita que atributos, identificadores ou toda a asserção seja cifrada. Tal característica permite garantir a confidencialidade fim a fim;
- **Perfils de atributo** – Simplifica a configuração e a implantação de sistemas que trocam dados de atributos. Define como os atributos poderão ser transportados nas asserções SAML. Define um perfil básico, que utiliza os tipos primitivos do XML para expressar os atributos e também define perfis como X.500/LDAP, UUID[7] e XACML;
- **Manutenção da sessão** – O SAML 2.0 provê um protocolo que permite que todas as sessões, providas por uma autoridade de sessão, possam ser facilmente encerradas simultaneamente;
- **Suporte a dispositivos móveis** – Se preocupa com as restrições de processamentos e largura de banda dos dispositivos;
- **Mecanismos de privacidade** – É possível expressar as configurações e políticas de privacidade dos provedores e principais, com relação ao uso da informação;
- **Descoberta do provedor de identidade** – Permite uma forma para localizar provedores de identidades, em ambientes onde exista mais de um provedor de identidade.

Nas versões 1.0 e 1.1 da SAML o principal objetivo era transpor domínios através do uso da autenticação única (SSO), possibilitando que usuários autenticados em um domínio de segurança pudessem usufruir dessa autenticação em serviços presentes em outros domínios, sendo isto transparente para o usuário. Era feito uso de uma identidade federada. Neste caso, a entidade *Provedor de Identidade* e o *Provedor de Serviços* entravam em acordo sobre os atributos dos usuário, como por exemplo, o nome do usuário e atributos de sessão, cabendo ao Provedor de Identidade garantir a autenticidade dos mesmos ao Provedor de Serviço. A figura 2.8(a) ilustra um caso de identidade federada.

Com a SAML 2.0 surgiu uma nova forma de uso de identidade federada, que permite a "*ligação entre contas*". Neste caso, as diferentes identidades de um usuário, presentes em diferentes Provedores de Serviço, podem ser associadas de forma que possibilite o SSO, porém sem ferir a privacidade do usuário. A SAML 2.0 propõe o uso de pseudônimos, o que evita que Provedores de Serviços entrem em acordo, visando rastrear as informações de um determinado usuário.

[7] Identificador único universal, definido pela OSF como parte do DCE, uma vez criado por alguém, tem-se a garantia que o mesmo não será reutilizado por mais ninguém [OpenGroup, 1997].



(a) Atributos federados (b) Ligação entre contas

Figura 2.8: Identidade federada

No caso apresentado na figura 2.8(b), o Provedor de Identidade criou diferentes pseudônimos com os Provedores de Serviço A e B, para referenciar um mesmo usuário, no caso João. Assim, João ao apresentar a asserção ao Provedor A, é reconhecido como o usuário local *ja456*; e ao apresentar a asserção ao Provedor B, é reconhecido como o usuário local *jb123*. Dessa forma, os provedores A e B, não terão meios para rastrear o usuário João.

WS-Security

A *WS-Security* [OASIS, 2004b] define um esquema XML, que provê meios para proteger as mensagens SOAP, permitindo a assinatura, a cifragem do cabeçalho, do corpo, ou de somente algumas partes da mensagem, empregando as especificações XMLDSign [Bartel et al., 2002] e XMLEnc [Imamura et al., 2002]. A flexibilidade é uma das características da especificação que permite que as mais variadas tecnologias de segurança, como por exemplo, Infra-estrutura de Chave Pública (ICP), Kerberos [Kohl e Neuman, 1993] e SSL [Freier et al., 1996] sejam utilizadas.

A *WS-Security* prevê o uso de mecanismos de segurança dentro das mensagens SOAP, utilizando o XMLDSign e o XMLEnc para a inclusão de assinaturas e cifragem. O caminho percorrido por uma mensagem SOAP, da origem até o destino final, pode ser composto por diversos nós SOAP intermediários. Neste caso, pode ser desejado que partes das mensagens sejam somente lidas por determinados nós ou ainda garantir que determinadas partes se mantenham íntegras até o destino final da mensagem e assim nenhum nó intermediário poderia modificar tal informação.

Visando atender tal caso, a especificação WS-Security permite a inclusão de múltiplas assinaturas e cifragens nas mensagens SOAP. Isto é feito através da inclusão de elementos XML `<wsse:Security>` nas mensagens. Cada elemento `<wsse:Security>` identifica, atráves do atributo `SOAP1.2:role`, o nó a qual aquela informação está direcionada, permitindo assim que aquele trecho da mensagem SOAP só será compreendido por um nó es-

pecífico. O elemento <wsse:Security> também pode ser usado para armazenar a assinatura do emissor inicial da mensagem SOAP, permitindo a todos os nós comprovem a origem da mensagem.

Cada nó intermediário só pode processar o elemento <wsse:Security> direcionado a ele, podendo assim removê-lo ou adicionar novos elementos <wsse:Security> antes de encaminhar para o próximo nó, presente no caminho da mensagem SOAP. É possível também que cada nó intermediário adicione novos subelementos a um elemento <wsse:Security> existente.

```
1  <soapenv:Envelope
2     xmlns:soapenv="..." xmlns:wsse="...">
3     <soapenv:Header>
4
5        <wsse:Security>
6           <wsse:UsernameToken wsu:Id="...">
7              <wsse:Username>joao</wsse:Username>
8           </wsse:UsernameToken>
9        </wsse:Security>
10
11    </soapenv:Header>
12    <soapenv:Body>
13       ...
14    </soapenv:Body>
15 </soapenv:Envelope>
```

Figura 2.9: Mensagem SOAP ilustrando o *WS-Security*

A figura 2.9 apresenta um exemplo de uma mensagem SOAP com o cabeçalho da *WS-Security*. O exemplo consiste em enviar uma simples credencial, no caso "joão" (linha 7), sem qualquer tipo de proteção. Na linha 2 da figura são informadas as URIs[8] para os espaços de nomes XML do SOAP e da WS-Security. Cada elemento <wsse:Security> (linhas 5–9) pode expressar informações sobre a cifragem, assinatura e sobre as credenciais de segurança. As linhas 6–8 expressam detalhes sobre uma credencial de segurança, porém os elementos <wsse:Security> podem conter mais de uma credencial de segurança, se desejado for.

A mensagem apresentada na figura 2.9 poderia ser empregada pelo seguinte cenário. Um nó SOAP, após receber uma invocação de um cliente que tenha se autenticado através de um mecanismo presente nas camadas subjacentes (como TLS/SSL), encaminha tal mensagem para outro nó SOAP, sendo que ambos nós estariam presentes dentro de um mesmo ambiente, considerado confiável e seguro. Assim, o objetivo da mensagem é indicar ao nó SOAP final que, em um nó SOAP mais externo, a autenticação do cliente já foi realizada e esta informação está sendo repassada através do elemento <wsse:Username> (linha 7). Assume-se também que a segurança da comunicação entre os nós SOAP é garantida através, por exemplo, do TLS/SSL. A figura 2.10 ilustra tal cenário.

[8] A URI foi suprimida para facilitar a visualização do código.

Atualmente a especificação provê suporte a dois tipos de credenciais de segurança: credenciais `UsernameToken` e credenciais `BinarySecurityToken`. Um uso para a credencial `UsernameToken` foi descrito anteriormente e apresentada na figura 2.10. Já a credencial `BinarySecurityToken` apresenta uma forma padrão para anexar a um pedido SOAP qualquer credencial de segurança codificada em forma binária, por exemplo, certificados X.509.

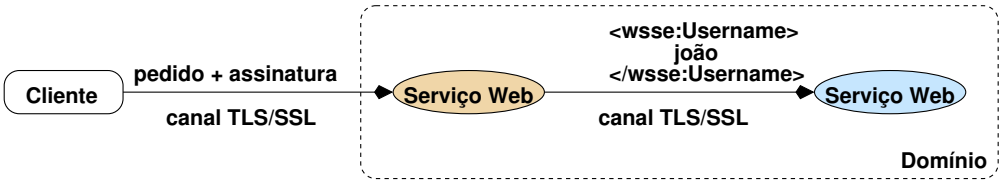


Figura 2.10: Encaminhando a identificação do cliente [Weerawarana et al., 2005]

No exemplo apresentado na figura 2.10 a confidencialidade e a integridade das mensagens era garantido através do uso do TLS/SSL, ou seja, na camada de transporte. Porém, poderia ser feito uso do XMLEnc e do XMLDSign para garantir tais propriedades, evitando assim a necessidade do TLS/SSL. Outra forma ainda seria a combinação do TLS/SSL com o XMLEnc e o XMLDSign.

WS-Trust

Desenvolvida por um conjunto de empresas, lideradas pela Microsoft e IBM, a *WS-Trust* [WS-Trust, 2005] é uma proposta que visa principalmente a troca de atributos de segurança através de diferentes domínios administrativos.

A *WS-Trust* baseia-se no caso em que os atributos de segurança, considerados válidos pelo provedor do serviço, já estejam contidos no pedido originado pelo cliente ao serviço. O serviço pode indicar, ao cliente, os atributos desejados através de sua política de segurança, descrita de acordo com a especificação *WS-Policy* [WS-Policy, 2004]. No caso do cliente não possuir os atributos desejados pelo serviço, este poderá requisitá-los às autoridades apropriadas, indicadas pela política do serviço.

O serviço de atributos de segurança (*Security Token Service* – STS) é definido pela *WS-Trust* como a autoridade responsável por emitir, renovar e validar os atributos de segurança, sendo este a base do modelo de confiança. O STS consiste de um Serviço *Web* que implementa uma interface WSDL, especificada pela *WS-Trust*, e processa mensagens SOAP seguras. A interface do STS define duas operações: `wstrust:RequestSecToken` para realizar o pedido; e `wstrust:RequestSecTokenResp` para a obtenção dos atributos de segurança.

A figura 2.11 ilustra um caso típico de confiança mediada através do STS. O cliente invoca o Serviço *Web*, entretanto, de acordo com a política de segurança deste serviço, é necessário que o cliente apresente credenciais de segurança emitidas pelo STS. Assim, o

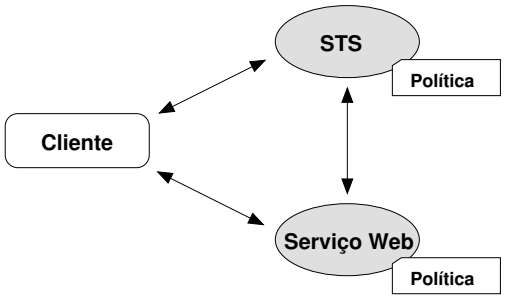


Figura 2.11: O uso do STS na mediação de confiança

cliente invoca a operação `wstrust:RequestSecToken` do STS a fim de obter as credenciais necessárias. O pedido ao STS pode conter credenciais que autentiquem o cliente ao STS. Uma vez que o STS tenha analisado as credenciais fornecidas pelo cliente e verificar que as mesmas cumprem os requisitos necessários, o STS responde com a mensagem `wstrust:RequestSecTokenResp`, que contém as credenciais necessárias para que o cliente consiga acessar o Serviço *Web*.

A autenticidade da resposta pode ser garantida através de assinaturas digitais e a resposta ainda pode conter informações adicionais, como tempo de vida da credencial e mecanismos para proteção contra ataque de mensagens antigas. Por fim, o cliente efetua um novo pedido ao Serviço *Web*, juntamente com as credenciais de segurança obtidas junto ao STS. O Serviço *Web* verifica as credenciais apresentadas e assim garante o acesso ao recurso.

A especificação da WS-Trust não se preocupa com o estabelecimento das relações de confiança, mas usufrui das relações já estabelecidas possibilitando assim, que partes que possuem relações estabelecidas, possam se comunicar. As relações de confiança podem ser construídas através de raízes fixas, onde é definido um conjunto fixo de entidades em quem se confia; através de confiança hierárquica, onde o confiança é dada através de uma árvore e os nós inferiores confiam nos nós superiores; ou ainda, através do uso das *redes de confiança*, onde cada entidade determina em quem confiar.

WS-Federation

A *WS-Federation* [WS-Federation, 2003], proposta liderada pela Microsoft e IBM, define mecanismos que permitem que diferentes domínios de segurança possam se federar e assim usufruir da mediação da confiança para o compartilhamento de identidades, atributos e autenticação entre os participantes.

Os modelos definidos nas especificações *WS-Security* [OASIS, 2004b], *WS-Policy* [WS-Policy, 2004] e *WS-Trust* [WS-Trust, 2005] formam a base para a federação de identidades. O STS, definido na *WS-Trust*, é um Serviço *Web* que emite atributos de segurança utilizando um conjunto padrão de mensagens. O STS também pode ser considerado como um *Provedor de Identidade* (*Identity Provider* – IdP) que ao receber, por exemplo, um pedido assinado,

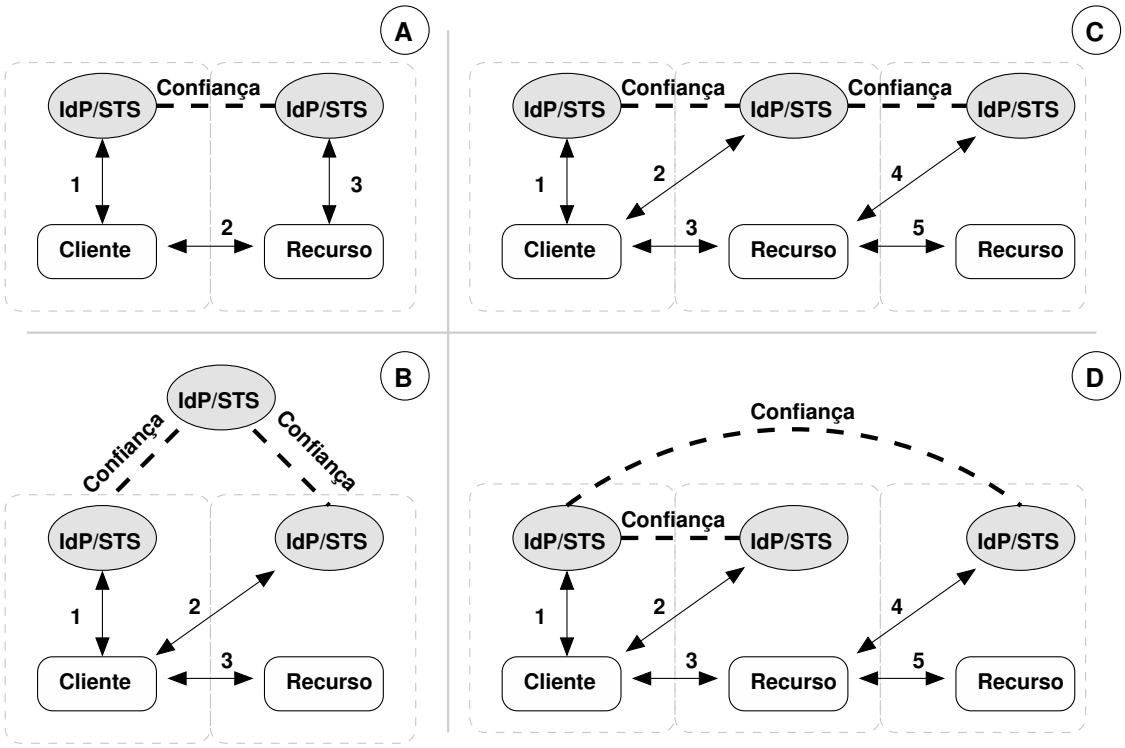


Figura 2.12: Alguns casos de mediação de confiança [WS-Federation, 2003]

emite uma credencial de segurança, garantindo a autenticação ou autorização de um cliente para os demais serviços presentes na relação de confiança. A *WS-Federation* apresenta diversas formas de mediação de confiança, ilustradas pela figura 2.12, contudo a especificação não indica como tais relações de confiança são estabelecidas.

No cenário "A" da figura 2.12, um cliente se autentica primeiramente no IdP/STS dentro do seu domínio e encaminha a prova dessa autenticação, juntamente com o pedido, ao recurso (passos 1 e 2). O recurso solicita ao teu IdP/STS para que o mesmo valide a credencial recebida e, por já existir uma relação de confiança entre os serviços IdP/STS dos dois domínios, a credencial é considerada válida (passo 3).

No cenário "B", a confiança entre os domínios do cliente e do recurso é mediada através de um outro IdP/STS, pois o mesmo possui relações de confiança com ambos domínios. No passo 1 o cliente se autentica no seu IdP/STS e solicita ao IdP/STS do recurso para que o mesmo valide a autenticação e lhe conceda os atributos de segurança necessários para acessar o recurso (passo 2). O fato de existir um caminho de confiança, intermediado por uma terceira parte confiável, entre os IdP/STS do cliente e do provedor de serviços, permite que asserções emitidas em um domínio sejam consideradas válidas em outro domínio.

O cenário "C" apresenta uma forma mais completa de interação entre os serviços. A proposta dos Serviços *Web* é permitir que aplicações interajam com outras aplicações, assim no passo 1 o cliente se autentica em seu IdP/STS e solicita ao IdP/STS do recurso os atributos necessários para acessá-lo (passo 2). Para atender a requisição do cliente, o provedor de

recursos necessita obter algumas outras informações, sendo estas providas por um outro provedor em um outro domínio. Assim, o provedor de recursos intermediário, fazendo uso da relação de confiança entre os domínios, invoca o Idp/STS do outro provedor de recurso (passo 4) para obter as credenciais necessárias para então acessar o recurso (passo 5).

Neste ponto, o provedor de recursos do domínio intermediário deseja obter alguma informação de um outro provedor de recurso e assim usufrui da confiança existente entre os domínios para solicitar ao IdP/STS do outro provedor de recurso (passo 4) os atributos necessários para obter acesso ao recurso (passo 5).

No cenário "D", o cliente após se autenticar em seu domínio (passo 1), solicita atributos junto ao IdP/STS do recurso (passo 2) e ao acessar o recurso (passo 3), delega também as credenciais necessárias para que este recurso possa solicitar ao IdP/STS do recurso final (passo 4), os atributos necessários para comunicar com o recurso (passo 5). A delegação foi necessária visto que o domínio do recurso intermediário não possui uma relação de confiança com o domínio final.

Em ambientes federados, um principal autentica-se uma única vez e usufrui dessa autenticação em todos os domínios por onde passa (SSO). Neste caso, um outro fator que deve ser considerado é quando a sessão desse principal termina. A especificação *WS-Federation* define uma forma para propagar uma notificação para todos os domínios presentes na transação do principal, informando o encerramento daquela sessão.

Outro ponto abordado pela *WS-Federation* é a propriedade de "privacidade". A autenticação única permite que o principal usufrua dos direitos, obtidos na autenticação inicial, em todos os domínios por onde exista uma relação de confiança. Sabendo que uma transação de negócios pode envolver diferentes serviços, seria possível, através de um cruzamento de dados, obter informações pessoais de um determinado principal. A *WS-Federation* define o "Serviço de Pseudônimos e Atributos" o que permite que um principal possua diferentes *apelidos* em diferentes domínios ou recursos, permitindo ainda que este *apelido* seja alterado em cada sessão.

2.3 Gerenciamento no ambiente dos Serviços Web

2.3.1 Gerenciamento de identidades

Uma identidade digital consiste na representação de uma entidade em um domínio específico e geralmente está relacionada a domínios do mundo real. Uma entidade pode possuir múltiplas identidades, sendo cada identidade constituída por um conjunto de características, podendo estas serem únicas ou não a um domínio.

O *gerenciamento de identidades* consiste de um sistema integrado de políticas, processos de negócios e tecnologias que permitem às organizações proverem recursos de forma

segura, somente aos seus usuários. Diversos modelos foram propostos para o gerenciamento de identidades e em [Jøsang e Pope, 2005; Jøsang et al., 2005a] é apresentada uma breve descrição de alguns modelos.

O ***modelo tradicional*** trata a identificação de forma isolada onde o provedor de serviço também atua como o provedor de identidades e de credenciais (senhas associadas com os identificadores). Neste modelo, os usuários possuem identificadores únicos e específicos para cada serviço com o qual interajam. E como conseqüência, diferentes credenciais associadas com cada identificador.

Com o crescimento da oferta de serviços, o gerenciamento de identidades digitais, por parte dos usuários, tornou-se uma tarefa árdua. Hoje é comum que uma pessoa possua diferentes identidades para interagir com o seu provedor de *e-mails*, com o sistema de uma empresa aérea, com um sítio de notícias, com o sítio de um supermercado, etc. Cada sistema exige um conjunto próprio de informações para que se possa criar uma identidade digital. Na maioria das vezes, os provedores de serviços solicitam as mesmas informações aos usuários, como nome, endereço, telefone, e principalmente, um identificador único e uma senha.

Para os usuários é muito custoso alimentar bases de dados de diferentes serviços, repetindo sempre as mesmas informações, entretanto a principal dificuldade é gerenciar o identificador e a senha escolhidos para cada sistema. Uma forma mais simplificada, porém não ideal, o usuário poderia sempre utilizar um mesmo identificador e senha para todos os sistemas. Mas na prática isso pode não ser possível, visto que o identificador escolhido por um usuário em um sistema, já possa ter sido escolhido por um outro usuário em um outro sistema.

O modelo de ***gerenciamento de identidades federadas*** surgiu para suprir as necessidades apresentadas pelo modelo de gerenciamento tradicional. Neste tipo de ambiente, é definido o conceito de **domínios**, nos quais estão presentes os provedores de serviço, de identidades e de credenciais relacionados, por exemplo, a uma determinada empresa. Assim, cada empresa constitui um domínio.

No ambiente de identidades federadas, acordos entre domínios permitem que identidades locais a um domínio sejam reconhecidas nos demais domínios participantes. Neste caso, é estabelecido o mapeamento dos identificadores de um usuário em diferentes domínios. Por exemplo, o identificador `joao.pedro@empresa` oriundo do domínio `empresa`, dentro do domínio `universidade` será mapeado para o identificador `jp@universidade`.

A federação de domínios de identificação, dá a impressão aos usuários de possuírem um identificador único para todos os domínios que compõem a federação. Os usuários poderão continuar a manter identificadores locais a cada serviço ou mesmo domínio, porém o simples fato de possuírem tal identificador, permite a estes usuários, acessarem serviços presentes em qualquer domínio da federação. O projeto *Liberty Alliance* e o projeto *Shibolleth*[9] são

[9] `http://shibboleth.internet2.edu`

implementações abertas de modelos de gerenciamento de identidade federada.

No ***modelo centralizado*** considera-se a existência de um único provedor de identidades e de credenciais em uma federação, o qual é utilizado por todos os provedores de serviços da mesma. Neste modelo um usuário pode acessar todos os serviços presentes na federação utilizando um mesmo identificador. Em tese o modelo se assemelha ao modelo de identidade federada, porém com a diferença de não necessitar do mapeamento de identificadores. A *WS-Federation* (ver seção 2.2.2) é um exemplo deste tipo de modelo. A *WS-Federation* especifica o *provedor de identidade* que tem o objetivo de autenticar os usuários, permitindo que estes usufruam desta autenticação em todos os serviços da federação.

Para os usuários a integração dos domínios, nos quais estão contidos os provedores de serviços, apresenta grandes benefícios. A facilidade de uma única autenticação (*Single Sign-On* – SSO), permite ao cliente efetuar o processo de autenticação uma única vez, seja em um provedor de serviço qualquer ou em uma entidade autenticadora centralizada, e usufruir deste processo de autenticação nos demais sistemas, identificados como parceiros de negócio. Para as empresas o compartilhamento de identidades digitais provê uma certa facilidade no gerenciamento dos usuários, visto que usuários de outras empresas não precisarão estar presentes em sua base para que possam interagir com o sistema.

Juntamente com a facilidade trazida pela autenticação única tem-se novos desafios. Do ponto de vista da segurança dos usuários deste sistema, a autenticação única permite, por exemplo, que provedores de serviço entrem em comum acordo para rastrear as atividades de um determinado usuário, ferindo assim sua privacidade. Já para os provedores de serviço, os novos desafios apresentados estão voltados para a gerência das relações de negócio entre os provedores parceiros. Visto que cada sistema participante do negócio possui suas próprias políticas de negócio, de segurança e administrativas. Por exemplo, como garantir que os controles de autenticação e de acesso aplicados em um domínio serão equivalentes aos controles aplicados em um outro domínio?

O *gerenciamento de identidade* surge como um ponto crítico para a viabilização de negócios neste novo cenário, objetivando a redução de custos, eficácia operacional e como conseqüência, um crescimento na quantidade negócios realizados. O gerenciamento de identidade também envolve aspectos relacionados com a definição, certificação e gerenciamento do ciclo de vida das identidades digitais, infra-estruturas para troca e validação dessas informações, juntamente com os aspectos legais.

Damiani et al. [2003] apresenta um estudo sobre os problemas inerentes ao gerenciamento de múltiplas identidades, descrevendo os requisitos necessários que um sistema de gerenciamento de identidades deve atender. Dentre os requisitos apresentados, alguns estão diretamente preocupados com as necessidades de segurança dos clientes, seguindo uma nova tendência [W3C, 2002; Rannenberg, 2000; Asokan et al., 1997], visto que geralmente os conceitos de segurança estão direcionados a proteção dos provedores de serviços, deixando de lado os requisitos de segurança dos clientes. Alguns dos requisitos apresentados são:

- *Privacidade* – Os clientes devem possuir meios para que possam expressar, e fazer valer, suas preferências de privacidade sobre informações relacionadas a sua identidade digital. No ambiente tradicional de comércio eletrônico, o provedor de serviços até apresenta um termo de responsabilidade, o qual indica a política de privacidade que rege aquela instituição, porém não existem mecanismos que possam garantir, para os clientes, que a política será realmente respeitada. Um grupo de trabalho da W3C está desenvolvendo uma especificação para que permitir que sítios *web* expressem suas políticas de privacidade, denominada *Platform for Privacy Preferences* (P3P) [W3C, 2002]. A especificação também descreve agentes presentes no lado dos clientes, que tem por objetivo obter de forma transparente ao cliente, as políticas de privacidade dos sítios e ainda tomar decisões, de forma automática, de acordo com o que foi verificado;

- *Anonimato* – Os clientes devem poder se manter anônimos, caso nenhuma informação sobre sua identidades seja requerida, garantindo que todas informações fornecidas não possam resultar na descoberta de sua identidade. O principal desafio neste caso, é como garantir que as informações fornecidas, de forma anônima, sejam confiáveis. Estão surgindo algumas soluções na literatura [Liberty, 2003; OASIS, 2005d; WS-Federation, 2003] para tratar do anonimato no ambiente dos Serviços *Web*;

- *Gerenciamento das identidades* – O sistema deve prover meios para que os clientes possam gerenciar as informações relacionadas às suas identidades. Deve-se ainda, prover formas para a revogação de identidades que não são mais utilizadas, a fim de evitar o uso indevido de credenciais expiradas, caracterizando um roubo de identidade;

- *Interoperabilidade das identidades* – No ambiente onde as identidades poderão atravessar múltiplos domínios administrativos e de segurança, é importante definir um formato para representar tais identidades, de maneira que possa ser compreendida por todas as entidades participantes de uma relação de negócio;

- *Responsabilidade* – O sistema deve garantir que todas as partes irão cumprir suas obrigações em uma transação de negócios. É necessário que exista uma trilha de auditoria segura, que garanta a responsabilização das partes, porém sem que isso venha ferir o anonimato das mesmas. As informações para auditoria devem estar disponíveis para todas as partes envolvidas no negócio, porém o sistema de gerenciamento deve evitar o uso dessas informações de forma indevida;

- *Gerenciamento de confiança* – A confiança estabelecida entre todas as partes presentes em uma federação, permite que a autenticação, e as possíveis credenciais de segurança associadas a essa, sejam aceitas por todas as entidades que fazem parte da federação. A gerência de confiança está relacionada a definição de meios que possam garantir os níveis de confiança para cada entidade presente no ambiente. Alguns trabalhos na literatura [Liberty, 2003; WS-Trust, 2005] apresentam mecanismos que usufruem das relações de confiança já estabelecidas e alguns apresentam uma especificação formal de

como deverá ser o nível de confiança em uma credencial, de acordo com o mecanismo de autenticação utilizado. Por exemplo, autenticação realizada através de mecanismos biométricos pode ser considerada mais confiável que a autenticação através de nome de usuário e senha [Conklin et al., 2004].

2.3.2 Gerenciamento de confiança

O princípio básico de segurança em um sistema computacional consiste em garantir os recursos somente aos sujeitos autorizados pela política de segurança do sistema. Nos modelos de autorização discricionários, o sistema de controle de acesso a um recurso, ao receber uma requisição de acesso, verifica *quem* está requisitando, qual o *recurso* que está sendo requisitado e, através de uma consulta a uma matriz de acesso, verifica quais os *direitos* que este *requerente* possui sobre o *recurso*. Segundo Blaze et al. [1999], apesar de tal tipo de solução ser amplamente adotada, esta não é a ideal para os sistemas atuais, devido a grande dinâmica destes ambientes.

Como visto na seção 2.3.1, organizações estão se agrupando com o intuito de prover facilidades para seus clientes. Provedores de serviços compartilham informações sobre seus clientes, o que permite aos clientes usufruir, por exemplo, do uso de uma única autenticação para acessar qualquer serviço que faça parte do grupo de organizações. Outro ponto positivo é que os clientes não necessitam mais fornecer sempre as mesmas informações quando forem acessar algum serviço. Em tais ambientes a autorização não pode se restringir simplesmente aos identificadores de usuários, diante da impossibilidade de um serviço conhecer todos os usuários existentes nos demais provedores de serviço. Assim, neste tipo de ambiente o ideal seria expressar a autorização confrontando requisitos e condições com as propriedades dos usuários.

A federação de identidades tem como ponto fundamental as relações de confiança entre provedores de serviços e clientes, e entre os próprios provedores de serviço. O fornecimento de informações pessoais de um cliente a um provedor de serviço só ocorre depois do cliente ter certeza que suas informações serão manipuladas de maneira correta. Por outro lado, o provedor de serviço só irá conceder ao cliente o acesso ao recurso, se o mesmo confiar nas informações fornecidas pelo cliente. O mesmo ocorre nas relações de confiança entre provedores de serviço. Tais relações permitem, por exemplo, que um usuário autenticado em um determinado provedor possa usufruir dos recursos providos por outro provedor, pois ambos possuem um acordo indicando o compartilhamento de recursos para os usuários de ambos provedores.

O termo confiança pode assumir diversos sentidos em uma aplicação distribuída. Em segurança o mais usual é como garantir que as informações foram enviadas por uma origem confiável. No caso, a preocupação geralmente restringe-se a garantir as propriedades de autenticidade e integridade das mensagens. Para tratar tal problema diversos modelos foram

propostos, como por exemplo, o X.509 [Housley et al., 2002], PGP [Zimmerman, 1994] e o SPKI/SDSI [Ellison et al., 1999; Rivest e Lampson, 1996].

O modelo X.509 está fundamentado sobre a confiança nas chaves privadas das Autoridades Certificadoras (ACs). A proposta inicial do X.509 apresentava uma topologia hierárquica, onde os nós inferiores da hierarquia confiam nos nós superiores. Porém, na versão 3 do X.509 [Housley et al., 2002] foram introduzidas algumas flexibilidades, como a criação de pontes entre as ACs (certificação cruzada), o que permite ainda a criação de teias de confiança, que apesar de não ser amplamente adotado, consegue transpor a estrutura hierárquica inicial. O PGP e o SPKI/SDSI são exemplos de modelos de confiança baseados nas teias de confiança. Não existe uma hierarquia, cada nó da rede indica em que deseja confiar. Trata-se de um modelo igualitário e independente de qualquer estrutura centralizadora.

Em uma aplicação distribuída a confiança não se restringe simplesmente em garantir as propriedades de autenticidade e integridade. As informações trocadas entre clientes e provedores de serviços possuem um certo valor e a manipulação indevida das mesmas pode acarretar em prejuízos para ambos os lados. Por exemplo, um cliente não gostaria de fornecer o número do cartão de crédito para qualquer provedor de serviço. A confiança entre clientes e provedores de serviço é algo que pode estar fundamentada, por exemplo, sobre uma base de reputações, a qual poderia indicar que um determinado provedor de serviços sempre honrou suas negociações.

As relações de confiança podem ser unidirecionais. Por exemplo, no caso do modelo X.509, os usuários confiam nas chaves das Autoridades Certificadoras (AC) mas o inverso geralmente não é verdade. Ou ainda, as relações de confiança podem ser bilaterais, exigindo assim que clientes e provedores de serviço forneçam algum tipo de informação para o estabelecimento da relação. Como estabelecer e gerenciar tais relações de confiança é um problema que tem sido destacado em diversos trabalhos na literatura [WS-Trust, 2005; WS-Federation, 2003; Jøsang et al., 2005b; Spantzel et al., 2005; Liberty, 2003; Shibboleth, 2005; Winslett et al., 2002; Skogsrud et al., 2003].

Em alguns trabalhos assume-se que o estabelecimento de confiança é um processo manual e exigem que diversos requisitos burocráticos sejam cumpridos antes de criar a relação de confiança, por exemplo, para entrar na hierarquia das ACs do X.509 é necessário cumprir um conjunto de requisitos, sendo alguns deles relacionados à segurança física do local onde estará armazenada a chave privada da AC. Outros trabalhos tratam a confiança de uma maneira mais dinâmica e volátil. Por exemplo, para uma determinado fluxo de negócios é necessário que diversos provedores de serviço se agrupem e, uma vez que o fluxo tenha sido cumprido, tal relação é desfeita.

2.4 Conclusões do capítulo

A Arquitetura Orientada a Serviço provê uma camada de abstração que permite que aplicações existentes possam ser encapsuladas e tratadas como serviços, disponibilizando assim as funções de negócio das organizações, sem que haja necessidade de reescrever as aplicações existentes.

A arquitetura dos Serviços *Web*, em seu núcleo, foi projetada de forma que permita a transferência de documentos XML utilizando protocolos padrões da Internet. Tal simplicidade implica também no uso de outras tecnologias para que se possa construir aplicações distribuídas complexas. Os Serviços *Web* combinam as características de execução das aplicações juntamente com as características de abstração da Internet. Pode se dizer que o grande sucesso das aplicações para Internet se dá devido ao alto nível de abstração o que permite garantir a interoperabilidade entre as mais diversas aplicações, sistemas operacionais e equipamentos. Os Serviços *Web* exploraram esse nível de abstração associando uma lógica de negócios.

Neste capítulo foram realizados estudos sobre diferentes aspectos significativos para o ambiente dos Serviços *Web*, sendo estes tratados nas formas de gerência de identidades e de confiança. Alguns dos trabalhos apresentados apresentavam soluções para o compartilhamento confiável de identidades, atributos de segurança entre diferentes domínios administrativos, possibilitando que transações inter-domínios sejam realizadas de forma interoperável e segura. O estudo sobre tal literatura permitiu conhecer alguns dos problemas relacionados aos Serviços *Web* ou ainda relacionados interação entre diferentes domínios, o que serviu de base para a motivação do modelo de segurança proposto neste tese e que é detalhado nos demais capítulos.

Capítulo 3

Relações de confiança e o uso de identidades federadas no ambiente dos Serviços *Web*

Neste capítulo é apresentado o modelo de segurança computacional para aplicações distribuídas baseadas na arquitetura dos Serviços *Web*. Tal modelo segue a abordagem com mecanismos de autenticação centralizados e de autorização descentralizados (veja seção 2.1.2). O objetivo geral deste capítulo é a explanação das entidades e os relacionamentos entre as mesmas, presentes no modelo. Os próximos capítulos detalham como as características descritas aqui serão providas.

3.1 Introdução

Utilizando padrões amplamente estabelecidos, os Serviços *Web* permitiram a integração de sistemas computacionais distribuídos e heterogêneos. Entretanto, a integração só é possível se mecanismos e modelos de segurança de cada sistema puderem interagir entre si. Isto é, recursos de um determinado sistema, regidos por uma política de segurança local, deveriam estar acessíveis aos clientes de um outro sistema, mesmo diante de tecnologias de segurança diferentes.

Nesta tese, sistemas computacionais são classificados em domínios administrativos e de segurança. O conceito de *domínios administrativos* ilustra um grupo de entidades, sejam estas clientes ou provedores de serviços, que está sujeito a administração de uma entidade comum, por exemplo, uma empresa ou universidade. O conceito de *domínios de segurança* descreve grupos de entidades que, dentro de um domínio administrativo, fazem uso de mecanismos de segurança comuns. Por exemplo, entidades que usam Infra-estrutura de Chave Pública (ICP) como mecanismo para autenticação de seus usuários, constituem um domínio

de segurança. Na prática domínios de segurança quase sempre coincidem com domínios administrativos.

Alguns trabalhos na literatura [WS-Trust, 2005; WS-Federation, 2003; Liberty, 2003; Shibboleth, 2005] fazem uso do conceito de federação de identidades para atingir a transposição dos mecanismos de autenticação e autorização. Isso possibilita que credenciais emitidas em um domínio administrativo possam ser reconhecidas e aceitas em outros domínios (veja a figura 2.12). Entretanto, parte-se do pressuposto que todos os domínios administrativos participantes utilizam as mesmas tecnologias de segurança. Estes trabalhos também consideram a existência de relações de confiança previamente estabelecidas, e de certo modo estáticas, não apresentando meios para que novas relações sejam estabelecidas de acordo com a necessidade de um certo fluxo de negócio.

O modelo apresentado neste capítulo visa permitir a transposição de credenciais de segurança por diversos domínios administrativos e de segurança, além de possibilitar o estabelecimento dinâmico da confiança. Para tal, são propostos serviços e entidades bem como a dinâmica entre estas para assim chegar na solução.

3.2 Aspectos estruturais de um domínio

O modelo de segurança computacional proposto nesta tese agrupa entidades como clientes e provedores de serviços de acordo com a tecnologia de segurança subjacente, caracterizando assim *domínios de segurança*. Para o controle e manutenção destes domínios, propõe-se uma entidade denominada Autoridade de Gerência do Domínio (AGD), a qual possui as seguintes atribuições:

- Prover meios para o controle da base de membros do domínio;
- Atuar como uma Terceira Parte Confiável (TPC) que, através da emissão e validação de asserções de segurança, possibilita a interação entre clientes e provedores de serviços;
- Estabelecer e gerenciar relações de confiança com outras AGDs;

A Autoridade de Gerência do Domínio (AGD) é constituída por um conjunto de interfaces de Serviços *Web* e é através destas interfaces que a AGD se faz disponível para membros de um domínio e para outras AGDs. As interfaces apresentam meios para acessar serviços para o controle de membros, emissão e validação de asserções, bem como para o estabelecimento e manutenção de relações de confiança entre AGDs. A figura 3.1 ilustra as entidades presentes em cada domínio e as relações entre domínios.

No modelo foi assumida a abordagem com autenticação centralizada e autorização descentralizada, pois tem-se um balanceamento entre a complexidade de administração e a

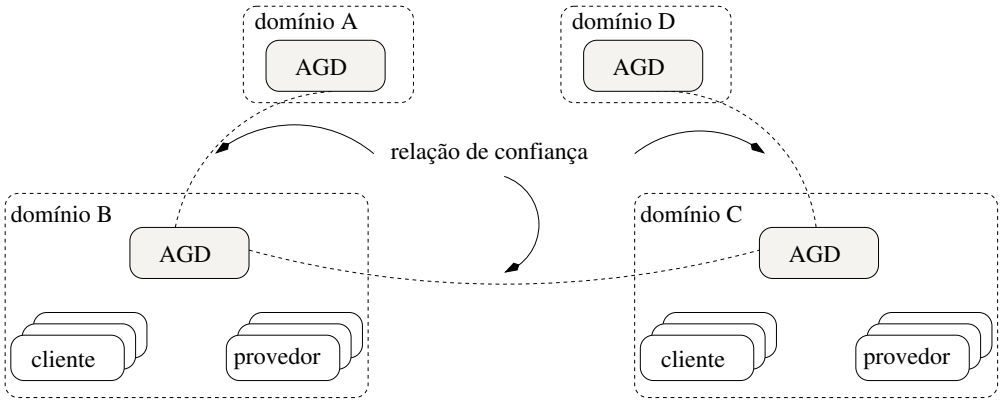


Figura 3.1: Entidades e relacionamentos de um domínio

adequação em ambientes de larga escala (ver seção 2.1.2). Dessa forma, membros de um domínio recorrem a sua AGD para que esta emita e valide asserções de segurança. Por exemplo, para que um cliente possa acessar um recurso de um determinado provedor de serviço, é necessário que este obtenha, junto à AGD, as asserções necessárias e as encaminhe ao provedor de serviço. Este por sua vez, pode consultar a AGD para verificar a validade dessas asserções.

Para a emissão e validação de asserções de segurança a AGD apresenta uma interface baseada no *Security Token Service* (STS) proposto em [WS-Trust, 2005] (ver seção 2.2.2). A escolha do STS como uma interface própria para lidar com asserções de segurança é devido à possibilidade de integração com outros sistemas baseados em Serviços *Web*. Um dos papéis do STS, que foi determinante para a sua escolha, é a troca de credenciais de segurança que permite ao modelo a transposição de credenciais entre diferentes domínios de segurança. O STS consiste apenas em uma interface padronizada para permitir a troca de asserções de autenticação, de atributos e de autorização. Tarefas como o controle de membros e a gerência da confiança entre os membros de um domínio são atribuições vinculadas somente a AGD introduzida no modelo. Desta maneira, clientes e provedores de serviços deverão se registrar e possuir uma relação de confiança estabelecida com a Autoridade de Gerência do Domínio (AGD) para que possam assim interagir com as demais entidades do modelo.

Em um único domínio, clientes fornecem as asserções de segurança, obtidas junto a AGD, aos provedores de serviços. O fato de ambos pertencerem a um mesmo domínio torna a validação dessas asserções uma tarefa simples. Em sistemas distribuídos compostos por diversos domínios administrativos, e de segurança, surge a necessidade de permitir que asserções emitidas em um domínio sejam consideradas válidas em outro domínio.

No presente modelo é feito uso do conceito de identidades federadas (ver seção 2.3.1) o que permite a transposição de credenciais de autenticação e autorização diante de diferentes domínios. Para tal, é assumido que as AGDs estabelecem relações de confiança entre si. Isso permite que um cliente de um domínio “A”, uma vez autenticado em seu domínio de origem, possa interagir com um provedor de serviços de um domínio “B”, sem que necessite fazer

uma nova autenticação[1]. As relações de confiança entre AGDs são estabelecidas seguindo o modelo de confiança não hierárquico, sendo que cada AGD possui autonomia para determinar com quais AGDs irá estabelecer relações de confiança. Tal modelo é semelhante àqueles apresentados pelo *Pretty Good Privacy* (PGP) [Zimmerman, 1994] e pelo SPKI/SDSI [Ellison et al., 1999; Rivest e Lampson, 1996] e evita a principal dificuldade apresentada pelos modelos de confiança hierárquicos, onde os nós inferiores da hierarquia devem expressar plena confiança sobre os nós que estão acima destes.

3.2.1 Atributos de segurança

A integração de diferentes domínios de segurança só é possível se as credenciais de segurança de um domínio puderem ser compreendidas em outro domínio. Com isso, surge a necessidade de uma linguagem comum para permitir tal integração, além de uma abordagem que permita correlacionar os direitos e requisitos entre domínios.

Em [OASIS, 2005c] foi apresentada a *Security Assertion Markup Language* (SAML) (ver seção 2.2.2), que consiste em uma forma para troca de asserções de segurança, essas escritas em XML. Asserções SAML podem indicar que um usuário foi autenticado e que possui um determinado conjunto de atributos, por exemplo, relacionados à aplicação. Entretanto, não provê meios para permitir o mapeamento de atributos entre domínios, além de não propiciar um local para a disponibilização de tais atributos.

Algumas experiências na literatura, como o *Shibboleth* [Shibboleth, 2005] definem um conjunto padrão de atributos visando, assim, garantir a interoperabilidade dentro de um ambiente federado e a disponibilização destes é feita através de uma Autoridade de Atributos. O projeto *Liberty Alliance* [Liberty, 2003] também propõe um Provedor de Atributos e a proposta *WS-Federation* [WS-Federation, 2003] faz o uso do UDDI [OASIS, 2004a] como um Serviço de Atributos que ficaria responsável por armazenar os atributos relacionados a usuários e provedores de serviço. Neste trabalho, a AGD é a entidade responsável por disponibilizar estas informações dos membros de seu domínio.

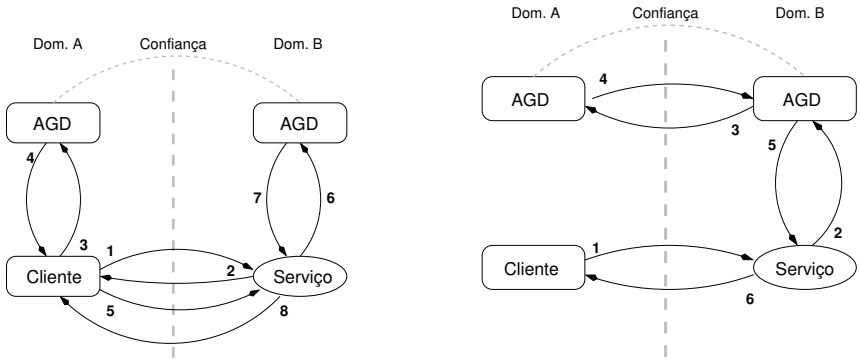
Em um ambiente federado a padronização dos atributos é tida como crucial [Shibboleth, 2005]. Para tratar tal problema, um conjunto padrão foi proposto com base nos documentos [Wahl, 1997; Smith, 2000; Internet2 e EduCause]. Aproximadamente 40 atributos foram definidos como *atributos de identidade comuns* [InComm], sendo que destes, 6 são altamente recomendáveis, 10 são sugeridos e 25 são opcionais. Nesta tese, foi realizado um estudo para a integração de duas diferentes tecnologias de segurança, o X.509 [Housley et al., 2002] e o SPKI/SDSI. Sendo assim, optamos por adotar um conjunto padrão de atributos formado pelos 6 atributos altamente recomendados e pelos 10 atributos tidos como sugeridos, como definido em [InComm]. Este conjunto é suficiente para permitir a transposição de credenciais

[1] Conceito conhecido como autenticação única (*Single Sign-On* – SSO).

de segurança de um domínio SPKI para certificados X.509. Entretanto, este conjunto pode ser expandido para acomodar outras tecnologias de segurança, como *tickets* Kerberos [Kohl e Neuman, 1993] ou credenciais biométricas.

Como visto na seção 3.2, o objetivo do modelo proposto aqui é permitir que clientes interajam com provedores de serviços, independente de quais domínios estes pertençam, desde que exista um caminho de confiança interligando os domínios de ambas as partes. Esta interação acarreta no fornecimento de credenciais de segurança do usuário perante o provedor de serviço, para que este último possa, por exemplo, garantir que o cliente em questão possui os direitos necessários para acessar o serviço.

Nesta tese a obtenção das credenciais de segurança pode seguir dois tipos de abordagens. Na primeira abordagem assume-se que o cliente é o responsável por fornecer os atributos necessários para acessar o recurso desejado. E em uma outra abordagem tal processo é transparente para os clientes, delegando a responsabilidade de obter os atributos do cliente aos provedores de serviço. A figura 3.2 ilustra estes dois tipos de abordagens.



(a) Cliente obtém os atributos (b) Serviço obtém os atributos

Figura 3.2: Obtenção de atributos

Na figura 3.2(a) o cliente tenta acessar um serviço (passo 1) que responde com um desafio indicando que o cliente necessita fornecer um determinado conjunto de atributos. Assim, nos passos 3 e 4 o cliente invoca sua AGD solicitando os atributos desejados pelo provedor do serviço. Tal dinâmica pode depender de interações com o usuário, por exemplo, o serviço solicita uma comprovação que o cliente realmente faz parte do domínio "A", mas a AGD só irá emitir tal atributo se o cliente se autenticar através do fornecimento de um nome de usuário e senha. No passo 5 o cliente fornece a credencial obtida com sua AGD ao provedor de serviço, o qual verifica a validade da mesma através de sua AGD (passos 6 e 7). Com a credencial validada, o serviço garante o recurso ao cliente (passo 8).

A abordagem apresentada na figura 3.2(b) busca ser o mais transparente possível para o cliente e para o provedor de serviço, ou seja, o cliente só precisará invocar o serviço e este por sua vez só precisará invocar a sua AGD para obter os atributos desejados. No exemplo, cliente e provedor de serviço estão em diferentes domínios e existe uma relação de confiança

entre as AGDs de ambos os domínios. Assim, o serviço ao receber a invocação do cliente (passo 1), invoca a sua AGD para obter os atributos (passo 2). Devido ao fato do serviço estar em um outro domínio, a AGD do serviço solicita a AGD do cliente os atributos desejados (passo 3). A AGD do cliente verifica se existe uma relação de confiança entre os domínios e se tudo estiver correto, envia tais atributos a AGD do serviço (passo 4). Por fim, a AGD retorna ao serviço os atributos do cliente (passo 5), que garante o acesso para o cliente (passo 6).

3.2.2 Transposição das credenciais de segurança

O uso de asserções SAML (ver seção 2.2.2), juntamente com as relações de confiança entre AGDs de diferentes domínios, permite a transposição das credenciais de segurança, isto é, um cliente que autenticou-se em seu domínio, pode usar a asserção dessa autenticação para acessar os recursos de um provedor de serviço em um outro domínio. Como dito na seção 3.2, clientes e provedores só estão aptos a trabalhar com a tecnologia de segurança de seus domínios e o fato de pertencerem a diferentes *domínios de segurança* requer que haja alguma forma de mapeamento das credenciais de um domínio para o outro. Em nossa proposta, a AGD além ser responsável pela emissão e validação de asserções de segurança para seus membros, através do serviço STS, também é responsável por fazer a tradução destas asserções para credenciais da tecnologia de segurança subjacente, possibilitando assim a transposição das credenciais de segurança por diferentes domínios de segurança.

A tradução de credenciais de segurança consiste na extração de informações de uma asserção SAML de autenticação para que então se possa compor uma nova credencial de autenticação, a qual poderá ser compreendida pelas demais entidades do domínio do servidor. A figura 3.3 ilustra o processo feito pela AGD, do provedor de serviços, para traduzir uma asserção SAML de autenticação em um certificado X.509, quando a mesma foi originada em um domínio SPKI/SDSI.

Em [OASIS, 2005a] são definidas diversas classes de contexto de autenticação que permitem à autoridade emissora da asserção de autenticação acrescentar informações adicionais sobre o processo de autenticação, como o mecanismo de autenticação empregado (usuário/senha, baseado em certificados SSL, assinaturas digitais). Neste trabalho foram utilizados somente os contextos para X.509 e SPKI. O primeiro indica que o cliente foi autenticado através de uma assinatura digital sendo que a chave deve ser validada como parte de uma ICP X.509. O contexto do SPKI indica que o cliente foi autenticado através de assinatura digital sendo que a chave foi validada através de uma rede de confiança do SPKI.

No exemplo apresentado na figura 3.3, o provedor de serviços faz parte de um domínio X.509, por conseqüência, só está apto a operar com certificados X.509 como meio para autenticação de outras entidades. Sendo assim, a AGD do provedor de serviços assume o papel de traduzir uma asserção SAML, emitida no contexto SPKI, para um certificado X.509,

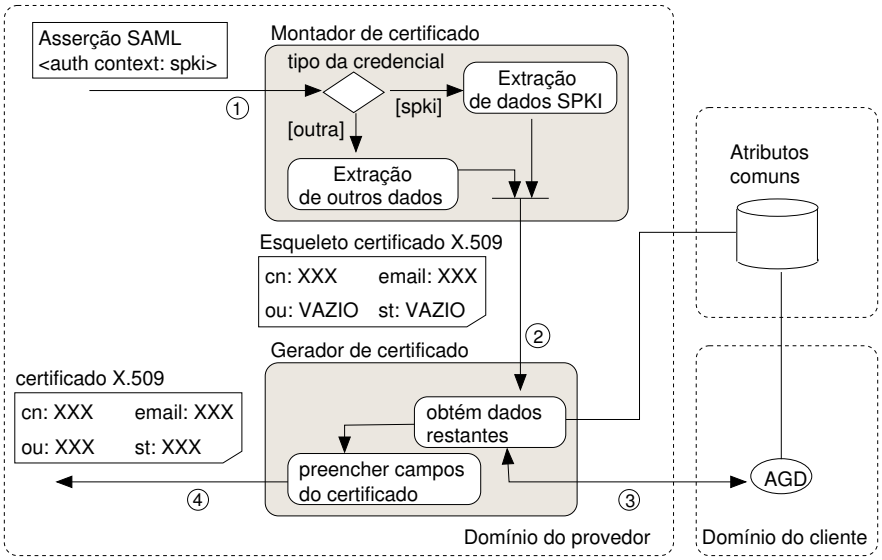


Figura 3.3: Tradução de credenciais de autenticação

auto-assinado. No passo 1 da figura 3.3 é fornecida uma asserção SAML de autenticação, indicando que esta foi criada no contexto SPKI, à AGD do provedor de serviço. O primeiro passo da tradução consiste em extrair da asserção atributos SPKI que possam ser mapeados diretamente para um certificado X.509, por exemplo, o nome comum e o endereço de *e-mail*.

Os atributos obtidos com a extração podem não contemplar todos os campos necessários para a criação de um certificado X.509 e cabe a AGD do provedor de serviços obter tais atributos invocando para isso a AGD do cliente (passo 3). Ambas AGDs compartilham um conjunto comum de atributos, conforme descrito anteriormente, o qual contempla as necessidades de ambos os domínios. Por fim, a AGD do provedor de serviços preenche os campos restantes e fornece ao provedor de serviço uma credencial de segurança que este compreenda (passo 4). Por se tratar de um certificado auto-assinado e pelo fato do provedor de serviço já expressar confiança em sua AGD, o certificado X.509 obtido pelo provedor de serviço é considerado válido sem que necessite recorrer a infra-estrutura do X.509 para fazer tal validação.

3.3 Classificação das relações de confiança

Todos os membros de um domínio expressam confiança sobre sua AGD, e essa por sua vez, expressa confiança em seus membros. Isto é, o fato de um cliente ou provedor de serviços ser membro do domínio indica que este cumpre todos os requisitos definidos pela AGD. No caso, existem relações de confiança entre a AGD e seus membros, mas o fato de clientes e provedores pertencerem a um mesmo domínio não implica que estes possuam confiança entre si. As relações de confiança entre entidades, clientes e provedores, presentes em um domínio, são arbitrárias e não se restringem aos limites do domínio. Cada entidade

está livre para expressar sua confiança com qualquer outra entidade pertencente ao mesmo ou a domínio diferente.

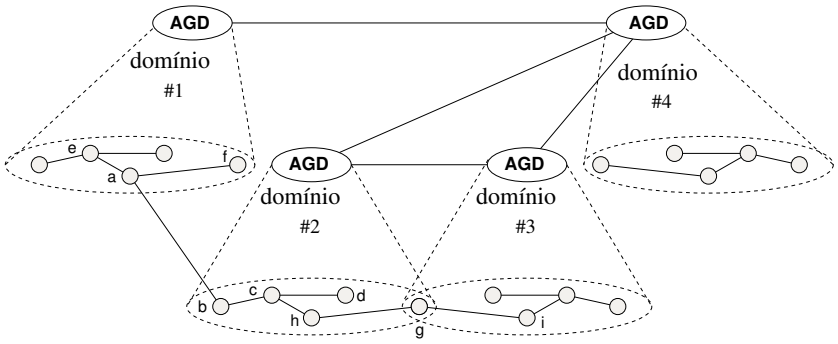


Figura 3.4: Relações de confiança: inter e intra-domínios

A figura 3.4 ilustra as relações de confiança entre entidades membros de domínios. A entidade "a", pertencente ao *domínio 1*, possui relações de confiança com três outras entidades, "e", "f" e "b"; sendo que a entidade "b" pertencente a um domínio diferente daquele de "a". E como em uma sociedade real, uma entidade pode pertencer a mais de um domínio. No cenário apresentado pela figura 3.4, a entidade "g" é membro dos domínios "2" e "3", e possui relações de confiança com a entidade "i" do *domínio 3* e com a entidade "h" do *domínio 2*.

Nos cenários de aplicações onde geralmente os Serviços *Web* são adotados, é comum casos onde existam diversos provedores fornecendo um mesmo tipo de serviço e cabe ao cliente determinar com qual destes provedores irá interagir. Por exemplo, algumas empresas de vendas de livros pela Internet estão provendo Serviços *Web* de forma que desenvolvedores de sistemas possam consultar as bases de livros dessas empresas. Para o desenvolvedor fica a tarefa em determinar quais desses Serviços *Web* ele deverá incluir em seu sistema, pressupondo que só é desejado a inclusão de um único serviço.

As relações de confiança entre membros de um domínio tem por finalidade representar as afinidades que um cliente ou provedor de serviços constroem ao longo do tempo. Estas relações são usadas na decisão sobre a escolha de um provedor de serviço, diante de diversos outros. As relações de confiança entre AGDs assumem dois papéis na presente tese. Além de permitir a autenticação única (SSO) aos membros dos domínios, auxiliam no estabelecimento de novas relações entre AGDs.

Como afirmado anteriormente, as relações de confiança entre membros em AGDs é binária, isto é, o fato de um cliente ou provedor ser membro, indica que este confia em sua AGD e vice-versa. As relações de confiança entre membros de domínios diferentes e entre suas AGDs assume uma abordagem difusa. Para cada relação é atribuído um peso o qual indica o quão forte é a relação em questão. Mais detalhes sobre como é feito tal ponderação e como esta pode ajudar na interação entre membros e entre AGDs são apresentados no capítulo 4.

Geralmente, a literatura trata o processo para o estabelecimento das relações de confiança entre domínios como estático que pode ser interpretado como composto de uma fase anterior que não é modelada. Por exemplo, administradores de duas AGDs encontram-se pessoalmente para firmar o contrato e após isto reconsideram suas políticas, administrativas e de segurança, indicando que a relação de confiança entre os dois domínios foi estabelecida.

Esse tipo de solução não é o ideal em ambientes dinâmicos, onde relações de confiança são estabelecidas de acordo com a necessidade de fluxos de negócio e são desfeitas logo após o término destes. Uma alternativa comumente utilizada para o estabelecimento dinâmico da confiança entre partes estranhas, consiste na confiança prévia de uma das partes perante a outra. Ou seja, uma parte está assumindo riscos, sem qualquer respaldo, ao expressar confiança em uma parte estranha, podendo isso acarretar em prejuízos.

O uso de uma Terceira Parte Confiável (TPC) surge como solução para o estabelecimento dinâmico da confiança. A TPC pode atuar como intermediária no processo de autenticação das partes, garantindo que cada parte realmente é quem diz ser, e também na lógica de negócios, garantindo que nenhuma das partes possa obter vantagens em detrimento da outra. Em modelos de confiança hierárquicos, como o modelo do X.509 [Housley et al., 2002], cada Autoridade Certificadora (AC) pode ser eleita como a TPC responsável por comprovar a identidade de cada uma das partes envolvidas na negociação.

No presente trabalho, o modelo confiança empregado pelas AGDs está baseado nas redes de confiança, sendo que qualquer entidade participante da rede determina em quem deseja confiar. Por não seguir um modelo de confiança hierárquico, como aquele presente no X.509, é necessário encontrar uma entidade que possa atuar com uma TPC, ou seja, é necessário encontrar um caminho de confiança[2] que interligue as duas partes em questão. Este caminho, quando composto por somente uma entidade intermediária, indica esta como sendo a TPC.

A realização de tais buscas por caminhos de confiança é algo que sempre foi negligenciado na literatura [Zimmerman, 1994; Ellison et al., 1999] e que surgiu de motivação para alguns outros trabalhos [Santin, 2004; de Mello et al., 2005, 2007]. No capítulo 5 é apresentada a nossa proposta para a localização de caminhos de confiança. O algoritmo proposto assume diferentes comportamentos com base nos pesos atribuídos a cada relação de confiança.

3.4 Implementação do protótipo

Com o intuito de validar o modelo proposto, principalmente o conceito de transposição de credenciais de segurança, optou-se pela implementação de um protótipo. Por se tratar de uma arquitetura de segurança a qual será utilizada por aplicações baseadas na arquitetura dos Serviços *Web*, o protótipo consiste de uma camada de segurança e de uma

[2]Um caminho de confiança consiste em uma ou mais entidades intermediárias que liguem duas entidades quaisquer.

aplicação exemplo, composta por clientes e provedores de serviços, que usufruem da camada da segurança nas interações entre si. Os códigos resultantes deste protótipo estão disponíveis na página do Grupo de Computação Segura e Confiável[3], dentro do contexto do projeto "Infra-estrutura de segurança para aplicações distribuídas orientadas a serviço" (CT-Info/MCT/CNPq 011/2005).

3.4.1 Portal de informações

Um caso interessante para o uso dos Serviços *Web* é o de **portal de informações**. Um portal tem por objetivo agregar informações provenientes de diferentes origens em uma única e simples interface, tornando-se um meio de fácil acesso para os usuários do sistema.

Os portais de informações passaram por diversas evoluções desde o seu surgimento na década de 90, quando se resumiam em diretórios e máquinas de busca para catalogar sítios *web*, até a mais atual versão que faz uso da tecnologia *Really Simple Syndication* (RSS) [RSS, 2005]. O uso do RSS trouxe facilidades para provedores de informações e principalmente para os portais, pois apresenta uma forma padronizada e simples para disponibilizar resumos de informações. Tal é tecnologia constituída de um serviço de publicação e assinatura de notícias, porém não permite aos portais agregadores de notícias uma maior interatividade com seus usuários.

Os Serviços *Web* podem ser utilizados para a construção de um mesmo de tipo portal que hoje faz uso do RSS. Os provedores de serviços disponibilizam uma interface padronizada para o fornecimento de informações para os portais e tais interfaces são publicadas em serviços como o UDDI [OASIS, 2004a]. Desta forma, os clientes dos portais recorrem ao serviço UDDI para localizar as informações desejadas, assinando assim os respectivos serviços. Com os Serviços *Web* é possível obter um nível maior de interação entre todas as entidades participantes, seja um cliente interagindo com os provedores de serviços, ou seja estes últimos interagindo entre si.

A aplicação exemplo apresentada aqui consiste em um portal de informações voltado para o entretenimento. Objetivo do portal é reunir em uma única interface diversos provedores de serviços que tenham como área de atuação o entretenimento pessoal, como por exemplo, cinemas, parques de diversão, vídeo locadoras, teatros, etc. O portal também reúne provedores de informações que não estão diretamente ligados ao entretenimento, mas que servem de base de apoio para a tomada de decisões dos usuários deste portal, como por exemplo, sítio de previsão do tempo, de resenhas de filmes, de companhias aéreas, de hotéis, de operadores de telefonia celular, etc.

A figura 3.5 ilustra os relacionamentos entre usuário, portal e provedores de serviço. Inicialmente, assim como a maioria dos serviços deste gênero, o portal exige um cadastro

[3] `http://gcseg.das.ufsc.br`

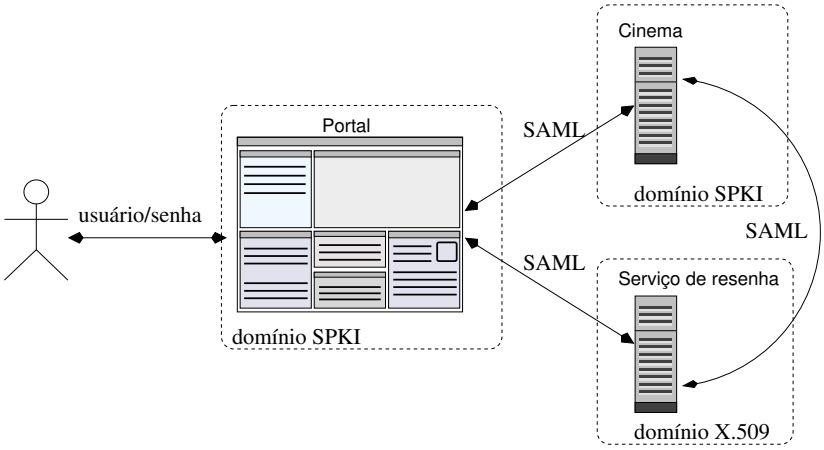


Figura 3.5: Portal de informações: interações entre serviços

por parte de seus usuários, para que estes forneçam suas informações pessoais como nome, endereço, idade, sexo, etc. Após esta etapa, um usuário pode selecionar os provedores de serviços de sua preferência e assim configurar sua página pessoal no portal. Se for o caso, o usuário pode ainda indicar suas preferências para cada serviço selecionado. Por exemplo, no serviço de cinema, um usuário poderá informar os gêneros de filmes preferidos, o melhor dia da semana e horário para ir ao cinema, etc.

Vale notar que os usuários autenticam-se com o portal através de um nome de usuário e senha e a propagação dessas informações para os demais serviços é feita através de asserções SAML. No exemplo apresentado pela figura 3.5, tem-se três domínios de segurança, cada qual gerenciado por sua AGD (veja seção 3.2). É tarefa da AGD emitir e validar asserções SAML para que os serviços de diferentes domínios possam interagir entre si, permitindo assim que o cliente possa acessar os recursos oferecidos pelos serviços de cinema e de resenha.

3.4.2 Ferramentas para implementação

Existem hoje diversas plataformas para o desenvolvimento de aplicações baseadas na arquitetura dos Serviços *Web*. Como é tradição deste grupo de pesquisa, optou-se pela escolha de ferramentas que seguem a filosofia do *software* livre, pois o acesso ao código fonte possibilita adequar as ferramentas às nossas necessidades. Assim, o protótipo foi construído basicamente sobre o servidor de aplicação Apache TomCat[4] e o Apache Axis[5], sendo ambas ferramentas desenvolvidas na linguagem Java. A arquitetura do Axis provê o conceito de interceptadores, denominados como *handlers* [Apache, 2005], que podem interceptar, de forma transparente, mensagens SOAP que são trocadas entre clientes e provedores de serviços. É com base neste conceito que a arquitetura de segurança aqui proposta é invocada, permitindo que asserções sejam emitidas e validadas de forma transparente para os desenvolvedores de aplicações.

[4] `http://tomcat.apache.org`
[5] `http://ws.apache.org/axis`

Outras bibliotecas de código aberto também foram usadas na implementação, sendo estas: XML-Security[6] que implementa as especificações *XML Encryption* [Imamura et al., 2002], cifragem de mensagens XML e *XML Signature* [Bartel et al., 2002], para a assinatura de mensagens XML; WSS4J[7] que implementa o serviço STS e provê o suporte para a troca de mensagens de acordo com a especificação WS-Security[OASIS, 2004b]; SunXACML[8] provê suporte ao modelo de controle de acesso XACML[OASIS, 2005b]; OpenSAML[9] para o suporte a asserções SAML[OASIS, 2005c].

A biblioteca WSS4J é capaz de lidar com a ICP X.509 como mecanismo para a emissão de asserções de autenticação, porém tal biblioteca não provê qualquer suporte para lidar com a ICP SPKI/SDSI. Para que pudéssemos representar um cenário com diferentes domínios de segurança, foram realizadas algumas extensões ao serviço STS e a biblioteca OpenSAML [Wangham et al., 2007], permitindo que a ICP SPKI pudesse ser utilizada para a emissão de credenciais de autenticação; e à biblioteca *XML Security*, possibilitando que elementos SPKI/SDSI pudessem ser inseridos em uma assinatura XML.

3.4.3 Dinâmica no protótipo

A figura 3.6 ilustra a dinâmica entre um cliente e um Serviço *Web*, sabendo que o cliente pode ser inclusive um outro Serviço *Web*. Os interceptadores ilustrados pela figura fazem parte da ferramenta Apache Axis e estes permitem que a camada de segurança aqui proposta ser acionada, de forma transparente, em toda troca de mensagens entre o cliente e o serviço.

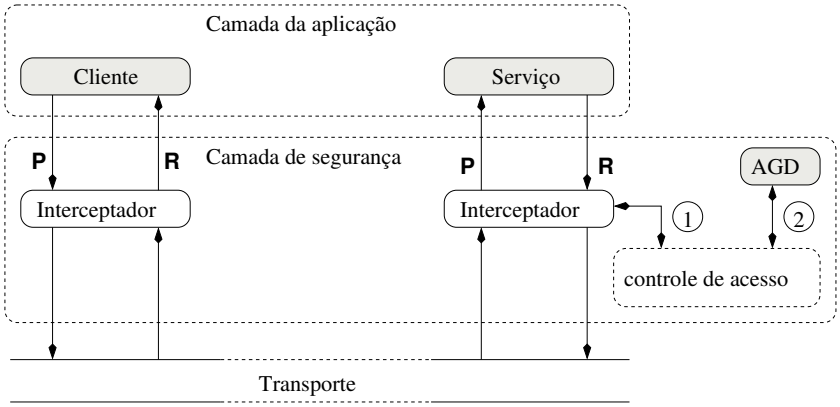


Figura 3.6: Dinâmica da aplicação

Os interceptadores podem assumir diversas tarefas, como cifrar e assinar mensagens XML, obter políticas de segurança, podendo ainda ser encadeados para atender uma determinada necessidade. Neste protótipo os interceptadores além de tratar a parte de cifragem e assinatura, também são usados para garantir a transposição das credenciais de segurança.

[6] http://xml.apache.org/security
[7] http://ws.apache.org/wss4j
[8] http://sunxacml.sourceforge.net
[9] http://www.opensaml.org

No cenário apresentado pela figura 3.6, o cliente faz uma invocação ao serviço, esta invocação é interceptada tanto no lado no cliente quanto no lado do serviço. O interceptador no lado do cliente faz, por exemplo, a cifragem e assinatura das mensagens de forma que somente o serviço possa decifrar além de validar a origem da mensagem. Juntamente com o pedido, o interceptador do cliente adiciona ainda asserções de segurança emitidas pela AGD do cliente.

No lado do serviço, o acesso ao recurso só é garantido ao cliente se este prover os atributos de segurança necessários. Assim, no passo 1 o interceptador invoca o módulo de *controle de acesso* o qual irá determinar se o cliente pode acessar o recurso desejado. O módulo de controle de acesso pode invocar a AGD (passo 2) caso as credenciais de segurança, fornecidas pelo cliente, sejam de uma tecnologia a qual este não compreenda. A AGD pode ainda ser invocada para obter outros atributos de segurança, cabendo a esta invocar a AGD do cliente para obtê-los (veja a figura 3.2(b)).

3.5 Conclusões do capítulo

Neste capítulo foi apresentado o modelo de segurança para aplicações distribuídas baseadas na Arquitetura Orientada a Serviço (AOS). Trata-se de um modelo com mecanismos de autenticação centralizados e de autorização descentralizados. A motivação para a concepção de tal modelo surgiu devido aos cenários nos quais os Serviços *Web* estão inseridos. Por se tratar de uma tecnologia integradora, que faz uso de padrões abertos, os Serviços *Web* são ideais para ambientes como a Internet, em que a interoperabilidade e o suporte para plataformas e redes heterogêneas são essenciais. Porém, a integração completa dessas aplicações distribuídas só é possível se mecanismos e modelos de segurança puderem interagir entre si, algo que ainda não foi bem abordado pela literatura.

O modelo aqui apresentado agrupa clientes e provedores de serviços em *domínios de segurança*, estes gerenciados pela entidade denominada Autoridade de Gerência do Domínio (AGD). No modelo a AGD possui diversas tarefas como controle de membros do domínio, a emissão e validação de asserções de segurança e o estabelecimento de relações com outras AGDs.

Diferentes domínios de segurança fazem uso de diferentes credenciais de segurança e a transposição destas só é possível se em algum momento houver o mapeamento de credenciais. O uso do SAML nos propiciou o transporte de asserções em uma língua franca[10] e juntamente com a proposição de um conjunto padrão de atributos (ver seção 3.2.1) foi possível transferir à AGD de cada domínio a tarefa de mapear asserções SAML em credenciais de segurança que são compreendidas pelos membros de seus domínios, por exemplo, em certificados SPKI/SDSI ou X.509, modelos que foram objeto de estudo neste trabalho.

[10]Uma língua que permite a comunicação entre grupos linguisticamente distintos

As relações de confiança assumem um papel importante no modelo e são tratadas em dois diferentes pontos de vista. A existência de uma relação de confiança entre duas AGDs permite que credenciais de segurança emitidas em um domínio sejam consideradas válidas pelo outro domínio. Isso traz facilidades aos clientes e provedores de serviços de ambos os domínios, permitindo aos clientes usufruírem do conceito da autenticação única (*Single Sign-On* – SSO) e tornando a administração de usuários menos custosa aos provedores de serviço.

As relações de confiança também ocorrem entre membros de domínios podendo inclusive transpor os limites dos domínios (ver seção 3.3). Tais relações servem de ajuda na tomada de decisões por parte dos clientes e provedores de serviço. Como visto no exemplo do *portal de informações* apresentado na seção 3.4.1, um cliente pode montar um portal o qual é composto por diversos provedores de serviço. Diante de um grande número de provedores de serviço como o cliente poderia escolher um determinado provedor de serviço diante de muitos outros? É neste ponto que o sistema de reputação, apresentado no capítulo 4, traz subsídios para esta tomada de decisão. Tal sistema de reputação depende e influencia as relações de confiança.

Tanto as relações de confiança entre AGDs e entre membros dos domínios são ponderadas de acordo com o sistema de reputação proposto no capítulo 4. A ponderação nas relações entre membros permite a estes determinar se as opiniões oriundas de um certo membro devem ou não ser consideradas, por exemplo, para determinar qual provedor de serviço utilizar para compor o portal de informações. Essa busca de opiniões realizada pelos membros pode ser propagada para o nível das AGDs, onde a ponderação daquelas relações possuem também a mesma finalidade.

As relações de confiança entre AGDs também podem ajudar no estabelecimento de novas relações de confiança entre partes estranhas. Assim, o modelo apresentado nesta tese visa o estabelecimento dinâmico da confiança, algo que não está bem coberto pela literatura. Por não se tratar de um modelo de confiança hierárquico, onde é possível encontrar facilmente uma entidade para intermediar o estabelecimento da confiança entre duas partes estranhas, surge a necessidade do emprego de algoritmos de busca para encontrar caminhos de confiança. No capítulo 5 é apresentado um estudo sobre algoritmos de busca para localização de caminhos de confiança que liguem duas partes estranhas.

Por fim, este capítulo teve por objetivo apresentar todas as facetas do modelo de segurança computacional proposto nesta tese, se atendo principalmente a transposição das credenciais de confiança e dando uma idéia superficial das demais partes abrangidas pelos capítulos conseguintes. Como resultados deste capítulo podemos destacar o desenvolvimento de um modelo para transposição de credenciais de segurança e a implementação de um protótipo para validar a eficácia do modelo.

Capítulo 4

Modelo de confiança combinado a um sistema de reputações

Neste capítulo é apresentada uma abordagem estatística, baseada em sistemas bayesianos, que compõe o núcleo de confiança destinado ao estabelecimento dinâmico da confiança entre entidades em ambientes complexos, como na AOS.

4.1 Introdução

No atual cenário da Arquitetura Orientada a Serviço (AOS) é comum existir inúmeros provedores de serviços oferecendo muitas vezes serviços similares e tratar a confiança em sistemas de larga escala tão e somente como garantir a identidade de uma entidade, seja esta um cliente ou um provedor de serviços, pode não ser suficiente, como é apresentado em alguns trabalhos [WS-Trust, 2005; Liberty, 2003].

Em [Gambetta, 1988; Sabater e Sierra, 2005] a confiança é entendida como uma probabilidade subjetiva que um indivíduo "A" espera que um indivíduo "B" execute uma dada ação na qual depende o bem-estar de "A". Nestes ambientes complexos caracterizados pela Internet, com interações momentâneas, surge então a necessidade de funções de gerenciamento de confiança que auxiliem, por exemplo, um cliente a decidir com qual provedor de serviço este deve interagir.

Segundo Khare e Rifkin [1998], inconsistências nas atuais relações de confiança em sistemas de larga escala indicam a necessidade de funções para um gerenciamento flexível da confiança, permitindo a navegação em complexas redes de confiança que se caracterizam por relações dinâmicas. Segundo Grandison e Sloman [2000] o gerenciamento de confiança diz respeito a coletar informações necessárias para estabelecer relações de confiança, avaliar os critérios relacionados, monitorar e reavaliar tais relações na evolução das interações.

O ponto crítico nestas redes de confiança é o estabelecimento inicial da confiança entre duas entidades que não se conhecem. Ambas estariam se arriscando em uma relação sem qualquer respaldo. As soluções apresentadas na literatura para tal problema consiste geralmente no uso de sistemas de reputações, através de provedores de opiniões, que torna possível avaliar a probabilidade da outra parte honrar uma negociação.

Esta probabilidade, ou a medida da confiança, é basicamente calculada de duas formas: através de médias ponderadas [Gray et al., 2003; Wang e Vassileva, 2003; Sabater e Sierra, 2001] ou através de métodos estatísticos [Buchegger e Boudec, 2003; Whitby et al., 2005; Teacy et al., 2006]. Os trabalhos que usam média ponderada normalmente atribuem um peso para cada provedor de opiniões (base de reputação), ou seja, as opiniões recebidas de provedores de opiniões considerados mais confiáveis terão uma maior influência no valor final sobre a confiança calculada para um dada entidade do que aquelas oriundas de provedores considerados menos confiáveis. Os métodos estatísticos consideram as interações passadas para prever como será o comportamento futuro, tanto dos provedores de opiniões quanto da entidade para qual se deseja formar um novo valor de confiança.

4.1.1 Motivação

A padronização dos Serviços *Web* permite que aplicações como portais de conteúdo sejam construídas de forma modular, possibilitando que diferentes provedores, que provenham serviços com soluções semelhantes, possam ser usados na composição do portal. Assim, cabe ao dono do portal, por exemplo, determinar qual provedor de serviço atende às suas necessidades de maneira mais apropriada. Se, por um lado, tal nível de personalização trouxe aos usuários finais facilidades em montar o portal, por outro lado, a grande oferta de provedores de serviços dificulta a escolha mais adequada daqueles que irão compor o portal.

Diversos fatores podem influenciar na decisão para a escolha de um determinado provedor de serviço diante de muitos outros, por exemplo, largura de banda disponível pelo provedor, o idioma do provedor e principalmente a confiança sobre tal provedor. De nada adianta obter informações rapidamente e no idioma desejado se não é possível garantir que as informações ali providas são corretas. Assim, a confiança existente entre um dono de portal e os inúmeros provedores de serviço irá determinar quais provedores deverão ser utilizados para compor o portal.

É neste cenário que modelo de confiança proposto neste capítulo busca propor uma solução. Como visto no capítulo 3, entidades presentes em uma AOS (clientes e provedores de serviço) são agrupadas em *domínios de segurança* e as interações entre tais entidades estão condicionadas à existência de relações de confiança entre estas. Na literatura, modelos de confiança estão sempre combinados com sistemas de reputação e a solução aqui proposta segue esta linha. Trata-se de modelo de confiança probabilista e a decisão de uma entidade confiar em uma outra entidade, e por conseqüência interagir com esta, será tomada com

base nas experiências passadas entre essas duas entidades. Na ausência de tais experiências, opiniões de outras entidades do sistema são usadas para que assim se possa formar um valor de confiança.

4.2 Confiança no ambiente composto por domínios

No capítulo anterior introduzimos o modelo o qual agrupou as entidades presentes na Arquitetura Orientada a Serviço (AOS) (ver seção 3.2) em *domínios de segurança*. Tal modelo implica no estabelecimento de relações de confiança entre os membros de um domínio com a Autoridade de Gerência do Domínio (AGD) e vice-versa, porém não implica que os membros de um mesmo domínio possuam relações de confiança entre si. Cada entidade (clientes, provedores de serviço e AGDs) possui autonomia para determinar com quais outras entidades irá estabelecer relações e quantificar a confiança de tais relações. Introduzimos nesta tese um sistema de reputação o qual é usado para o estabelecimento de qualquer relação de confiança. A idéia por trás deste sistema é que as relações de confiança já estabelecidas alimentam bases de reputações que auxiliam no estabelecimento de novas relações de confiança entre entidades.

O sistema de reputação é constituído de forma distribuída, isto é, não existe uma base de reputação centralizada a qual todas entidades do sistema alimentam e utilizam. Cada entidade membro possui uma *base de conhecimento* individual que agrega informações oriundas de experiências diretas e opiniões providas por outras entidades, sejam estas recebidas de outros membros ou de AGDs. Tal abordagem permite à cada entidade no sistema ter uma visão particular da rede de confiança e do sistema de reputações. Isto garante que cada entidade poderá expressar sua vontade e não irá depender de uma base única e comum a todas entidades, a qual pode não representar seu ponto de vista particular.

Apesar da *base de conhecimento* ser individual à cada entidade, a sua disponibilidade se dá por meio das AGDs. Isso permite retirar dos membros do domínio a complexidade em gerenciar tais bases, porém sem que impeça a visão particular que cada entidade possui do sistema de reputações. Como visto na figura 3.1, clientes e provedores de serviço podem pertencer a mais de um domínio e para estes casos, a *base de conhecimento* de tais entidades ficará replicada por cada uma das AGDs dos domínios a que estas entidades pertençam. Isso se justifica porque as AGDs também fornecem opiniões aos seus membros e a outras AGDs com quem possuam relações de confiança, sejam estas seus membros ou não. Diferentemente dos membros que fornecem opiniões com base em suas experiências anteriores, as AGDs utilizam as bases de experiências de seus membros para fornecer tais opiniões, pois nesta proposta as AGDs não possuem papel ativo, ou seja, estas não interagem com outros membros de forma que mantenham uma base própria de experiência direta.

O serviço de opiniões provido pela Autoridade de Gerência do Domínio (AGD) é inte-

ressante para casos onde uma entidade *i* deseja conhecer a reputação de uma entidade *j* e entre as suas relações não encontra qualquer informação a respeito da entidade desejada (*j*). Neste caso, a entidade *i* requisita, a sua AGD, opiniões sobre *j* e esta irá combinar todas as experiências que seus membros tiveram para então encaminhar para *i* um valor já formado. A figura 4.1 ilustra as bases de experiências dos membros (A, B, C e D) dispostas na AGD. Cada membro registra em sua base os resultados das interações que teve com outras entidades.

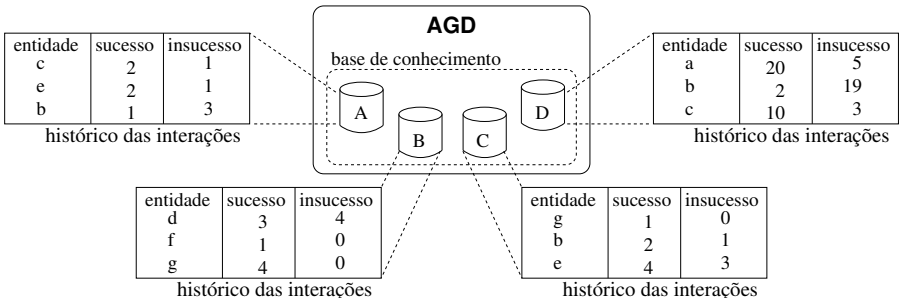


Figura 4.1: Bases de experiências dos membros dispostas na AGD

O uso de opiniões de membros oriundas de um único domínio não garante a visibilidade global. Por exemplo, um membro do domínio *X* deseja consultar a reputação da entidade *j*, que faz parte do domínio *Y*. Ao consultar a AGD do domínio *X*, o requisitante de informações constata que nenhum outro membro de seu domínio possui opinião sobre a entidade *j*. Para este caso, as relações de confiança entre AGDs permitem que uma AGD propague a consulta pelas demais AGDs, com quem possuam relações estabelecidas, garantindo assim a visibilidade global das informações sobre reputações.

4.2.1 Sistema de confiança e reputação

A interação entre um cliente *i* com um provedor de serviços *j* está condicionada a probabilidade deste último honrar a negociação. Tal probabilidade é calculada com base nas experiências diretas entre *i* e *j*, realizadas anteriormente, e no caso da ausência destas experiências, *i* poderá requisitar opiniões de outras entidades as quais possuem alguma informação acerca de *j*, inclusive das AGDs dos domínios nos quais o requerente pertença.

Como nos demais trabalhos na literatura [Buchegger e Boudec, 2003; Whitby et al., 2005; Teacy et al., 2006], optou-se neste trabalho por uma análise bayesiana para determinar a probabilidade de uma entidade *j* honrar futuras interações. Nos sistemas bayesianos para determinar a probabilidade (*a posteriori*) de *j* honrar as futuras interações é necessário conhecer a probabilidade *a priori* e esta pode ser obtida através de uma *função densidade de probabilidade* de uma *distribuição beta*[1] (equação 4.1).

[1]Uma *distribuição beta*, determinada pelos parâmetros α e β, é usada para representar variáveis aleatórias que são limitadas dentro de um intervalo, por exemplo, entre 0 e 1 [Jain, 1991].

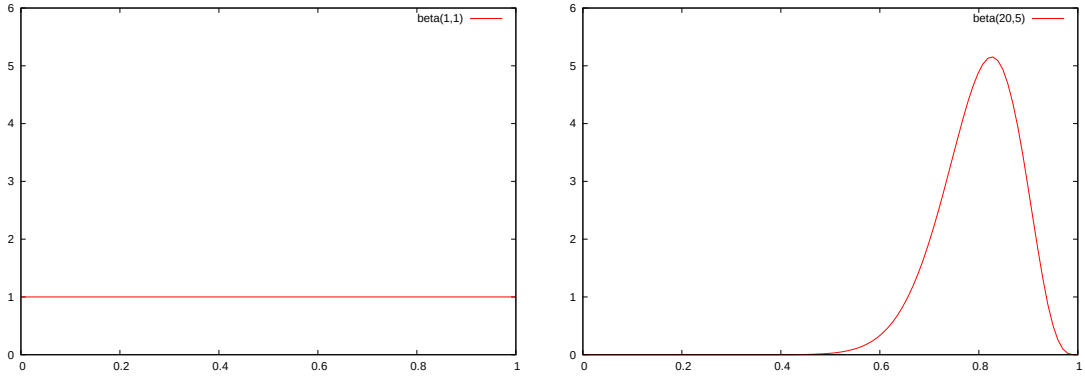
$$f(p) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \text{ sendo } \alpha,\beta > 0 \tag{4.1}$$

$$\Gamma(x) = \int_0^\infty e^{-t}t^{x-1}dt \tag{4.2}$$

sendo $\Gamma(x)$ a função Gama de Euler que estende a noção de fatorial para valores não inteiros.

O fato de só precisar de dois parâmetros (α e β), que são atualizados de forma contínua, torna a *distribuição beta* atrativa em sistemas bayesianos para calcular a probabilidade de uma entidade honrar uma negociação. Como apresentado na literatura [Buchegger e Boudec, 2003], os parâmetros α e β são utilizados como probabilidade *a priori* e em nosso modelo estes parâmetros funcionam como registros do total de interações entre um cliente i e um provedor j que resultaram em sucesso e insucesso, respectivamente.

Sistemas de reputações tornam-se mais precisos a partir do momento que suas bases possuam grandes quantidades de registros. Porém, em seu início a base de informação é nula. Considerando o exemplo anterior, que a entidade i deseja calcular a probabilidade da entidade j honrar a próxima interação e sabendo que não existe qualquer experiência anterior entre i e j, tem-se os seguintes valores para os parâmetros de registro de i: $\alpha = totalSucesso + 1$ e $\beta = totalInsucesso + 1$. Assim, no início é assumido uma distribuição uniforme sendo que a probabilidade de qualquer interação ocorrer com sucesso ou sem sucesso é exatamente igual, isto é, a probabilidade de j honrar ou de não honrar uma requisição de serviço de i será exatamente igual (veja a figura 4.2(a)).



(a) distribuição beta(1,1) (b) distribuição beta(20,5)

Figura 4.2: Distribuição beta

A partir do momento que novas observações são realizadas por i sobre as interações com j, os valores de α e β são atualizados, o que resultará em uma probabilidade *posteriori* mais expressiva. A figura 4.2(b) ilustra um caso onde houve 19 interações que resultaram em sucesso e 4 que resultaram em insucesso ($\alpha = 19+1$ e $\beta = 4+1$). Por fim, a probabilidade p de j honrar a interação é calculada através do *valor esperado* $E(p)$ da distribuição beta (equação 4.3), que para o caso da figura 4.2(b) é igual a $0,8$. Uma vez calculada a probabili-

dade de j honrar a interação, cabe ao cliente i determinar, subjetivamente, se o valor obtido é suficiente para que possa interagir com j.

$$E(p) = \frac{\alpha}{\alpha+\beta} \tag{4.3}$$

Determinar o grau de confiança sobre uma entidade consiste também em considerar o contexto no qual tal entidade está inserido. A delimitação do contexto serve para expressar a confiança com uma maior precisão, haja visto que uma entidade pode atuar em diferentes contextos e para cada um destes a mesma pode assumir diferentes comportamentos. Por exemplo, um provedor de serviços oferece um serviço para previsões do tempo e um serviço para indicar as melhores empresas na bolsa de valores para se investir. Das diversas interações com o provedor foi observado que as previsões do tempo fornecidas sempre foram precisas, porém as indicações sobre empresas nem sempre resultaram em sucesso. Isso deixa o provedor de serviço com uma boa reputação no contexto de "previsões do tempo", entretanto o mesmo não teria boa reputação no contexto de "investimentos financeiros".

Interações com j		
contexto	sucesso	insucesso
0	10	1
1	5	1
2	4	5
3	3	1

Tabela 4.1: Histórico das experiências que i teve com j em diferentes contextos

No modelo de confiança proposto aqui, o contexto é tratado da seguinte forma. Cada entidade possui uma base de experiências diretas que contém os históricos das interações que realizou com as demais entidades (veja a tabela 4.1). Para cada entidade presente nesta base são registradas as experiências separadas por contexto $(s_0, \ldots, s_n)$. Essa separação permite determinar em quais contextos a entidade, para a qual se está calculando a confiança, se mostrou mais correta e no caso de não houver ainda qualquer experiência em um determinado contexto, a combinação de todas as experiências, não importando o contexto, pode ajudar a prever o comportamento para o contexto desejado. As interações seguintes, dentro do contexto desejado, iriam então aprimorar esta visão inicial da confiança.

4.2.2 Fontes de reputações

Para o cliente i determinar a probabilidade do provedor de serviços j honrar uma futura interação, deve-se inicialmente analisar as experiências anteriores entre os mesmos. Sabe-se que nem sempre é possível assumir que existem experiências diretas e uma forma de atacar este problema é através de sistemas de reputações. Na presente proposta a base de reputações

consultada por *i* envolve informações providas por entidades com quem *i* possui relações de confiança estabelecidas; e pelas Autoridades de Gerência do Domínio com as quais *i* se relaciona. Estas bases constituem o que chamamos de *base de conhecimento* da entidade *i*.

Um cliente *i* que deseja formar um valor de confiança para o provedor de serviço *j*, consulta sua base de relações, questionando a respeito de opiniões sobre *j*. Cada entidade *k* entre as relações do cliente *i* fornecerá então o *valor esperado* (veja a equação 4.3) obtido sobre *j*. Com esses valores em mãos, o cliente *i* agrega-os através da equação 4.4.

$$c_{ij}^{s} = \frac{\sum_{k}^{y} co_{ik}^{s} \times op_{kj}^{s}}{\sum_{k}^{y} co_{ik}^{s}} \tag{4.4}$$

sendo c_{ij}^{s} o valor de confiança formado por *i* sobre o provedor de serviços *j* no contexto *s*. co_{ik}^{s} representa a confiança que *i* possui sobre o provedor de opiniões *k* no contexto *s*. op_{kj}^{s} expressa a opinião de *k* sobre *j* no contexto *s*. Assim, a equação 4.4 mostra que as opiniões recebidas por *i* serão ponderadas de acordo com a confiança que *i* possui sobre os provedores de opiniões *k*. Por fim, cabe a *i* determinar, subjetivamente, se o valor c_{ij}^{s} é suficiente para que possa interagir com o provedor de serviços *j*.

Como descrito na seção 4.2.1, as AGDs também atuam como provedores de opiniões, entretanto as opiniões geradas por estas entidades consistem apenas na combinação das experiências que seus membros tiveram com a entidade para quem se deseja conhecer a reputação. Como ilustrado pela figura 4.1, a AGD detém as bases de conhecimento de seus membros, cada qual indicando a quantidade de interações que ocorreram com sucesso e com insucesso em um determinado contexto (p.e. previsão do tempo, etc.).

Dessa forma, a opinião de uma AGD sobre uma entidade *j*, consiste no valor esperado de uma distribuição beta com $\alpha = \sum_{l}^{m} a_{lj}^{s}$ e $\beta = \sum_{l}^{m} b_{lj}^{s}$, sendo a_{lj}^{s} o total de experiências com sucesso que o membro *l* teve com a entidade *j* dentro do contexto *s* e b_{lj}^{s} o total de insucessos. *m* é o número total de membros do domínio em questão que interagiram com *j*.

A opinião recebida de uma AGD, ou de um conjunto de AGDs, é agregada da mesma forma apresentada pela equação 4.4, o que permite a entidade que está requisitando opiniões determinar quais domínios apresentaram opiniões mais precisas no decorrer do tempo. Vale ressaltar que a entidade requisitante só poderá obter a opinião das AGDs daqueles domínios nos quais esta entidade é membro. Porém, as AGDs, para a composição de suas opiniões, podem requisitar opiniões a outras AGDs com quem possuam relações, garantindo assim que a busca seja propagada pela rede de confiança.

O provimento de opiniões por parte das AGDs pode ser vantajoso pois estas reúnem uma quantidade maior de entidades provedoras de opiniões. Vale ressaltar que a AGD somente agrega as opiniões de todos os seus membros, e de seus associados, e fornece essa opinião

agregada sem qualquer tipo de ponderação. Assim, cabe a entidade requisitante ponderar as opiniões fornecidas por cada AGD.

4.3 Serviços agregados a AGD

A Autoridade de Gerência do Domínio (AGD) além de agregar funções providas pelo *Security Token Service* (STS) [WS-Trust, 2005] (veja seção 3.2), como a emissão e validação de asserções de segurança, apresenta duas novas interfaces de serviços para permitir o registro e busca de opiniões, sendo estes: Serviço de Registro de Opiniões (SRO) e Serviço Agregador de Opiniões (SAO).

O Serviço de Registro de Opiniões (SRO) é utilizado pelos membros dos domínios para que estes possam manipular suas bases de experiências e para que possam buscar as opiniões de outros membros, com os quais possuam relações de confiança estabelecidas. Os membros de um domínio podem invocar o SRO sempre que realizarem interações com outras entidades no sistema, informando a entidade com quem interagiu, o contexto no qual interagiu e, por fim, indicando se a interação ocorreu com sucesso ou com insucesso.

O SRO só permite que os membros adicionem novas informações às suas bases de conhecimento e operações como modificação e exclusão destas informações não são permitidas. Ou seja, um membro só pode incrementar o total de casos de sucessos e insucessos sobre uma outra entidade. Esse comportamento previne que um membro manipule sua própria base de conhecimento de forma que o SRO forneça diferentes opiniões para diferentes requerentes. Por exemplo, uma entidade maliciosa m apesar de ter registrado que realizou n interações com sucesso com uma entidade e, consegue de alguma forma prever que uma entidade t, com quem também possui uma relação de confiança, pretende realizar negócios com e ou com f (seu amigo). A entidade m poderia então modificar sua base de forma que t pense que e não honrou as negociações com m, privilegiando assim f. Por fim, após tal consulta, m poderia modificar sua base de conhecimento (disponível através da AGD) de forma a não provocar suspeita e se e a vier consultá-la, irá encontrar informações que de fato ocorreram, ou seja, m indicará que e honrou as n negociações.

Dessa forma, o maior problema que um membro pode causar ao sistema de reputações está no registro de interações de forma maliciosa, por exemplo, registrar que uma interação ocorreu com insucesso, mesmo que na verdade ela tenha sido realizada com sucesso. A base de conhecimento desta entidade, apesar de ser incoerente, será a mesma para qualquer entidade que a consulte em qualquer momento e o sistema de reputação apresentado na seção 4.2.1 se encarregará de filtrar opiniões de entidades maliciosas.

O Serviço Agregador de Opiniões (SAO) é responsável por calcular a reputação sobre uma dada entidade e fornecer o resultado a entidade que o está invocando, podendo ser esta um membro do domínio ou mesmo uma outra AGD. O SAO pode ser invocado em três

casos: quando um membro deseja verificar o resultado das experiências diretas que teve com uma dada entidade; quando um membro deseja verificar as opiniões das outras entidades, com quem possui relações de confiança estabelecida, sobre uma dada entidade; ou quando o membro deseja verificar a reputação de uma dada entidade através de diversos domínios.

Por exemplo, para que um cliente i interaja com um provedor j é desejável que a primeira conheça a probabilidade de j honrar essa negociação. Assim, i poderia recorrer inicialmente a sua base de experiências diretas e verificar o resultado das interações que tivera com j. Neste caso, o cliente i invoca o SAO de seu domínio para que este faça tal cálculo (equação 4.3), uma vez que a base de conhecimento de i encontra-se em sua AGD. Caso o requerente considere que o valor obtido não seja suficiente para interagir com j, este pode então requisitar ao SAO opiniões sobre outras entidades em quem i confia (equação 4.4). Por fim, caso o resultado obtido ainda não seja o suficiente, i pode requisitar ao SAO para que este obtenha opiniões de outras AGDs acerca de j. Estas AGDs são aquelas com as quais a AGD de i possui relações de confiança diretas ou indiretas.

4.4 Experimentos

Para verificar a efetividade do modelo de confiança proposto foram realizados experimentos em um ambiente simulado. A dinâmica das simulações consistiu primeiramente em escolher uma entidade de forma aleatória, denominada "consultora" e a partir desta escolher um conjunto de entidades, denominadas "candidatas", com as quais a entidade consultora pretende interagir.

Cada entidade candidata possui um comportamento próprio o qual indica como irá se portar diante de cada interação que esta venha a realizar. O comportamento de cada entidade foi escolhido de forma aleatória e este é representado por um número real dentro do intervalo de 0 até 1. Entidades que tenham o $comportamento = 1$ irão honrar todas as interações e o $comportamento = 0$ indica que não irão honrar qualquer interação. Os valores intermediários são expressos a uma granularidade de $0,1$.

A interação entre a entidade consultora com uma entidade candidata está condicionada ao valor de confiança (veja seção 4.2.1) calculado pela consultora sobre a candidata. A interação só ocorrerá se o valor de confiança obtido for superior ao limiar definido pela consultora. Esse limiar é definido de forma subjetiva para cada entidade e em nossas simulações foi adotado o limiar de $0,5$. Por fim, para cada cenário simulado foi contabilizado o total de interações que resultaram em sucesso e em insucesso. A tabela 4.2 descreve os cenários utilizados nas simulações.

O estabelecimento do vínculo entre membros e seus gerentes de federações e o agrupamento de federações constitui uma rede de confiança e esta por sua vez pode ser vista como um grafo, sendo que as entidades, membros ou AGDs, representam os vértices e os

Cenários	Descrição
1	A entidade consultora irá interagir com as entidades candidatas sem consultar qualquer base de reputações, não usando sequer uma base própria de reputação.
2	A entidade consultora irá armazenar um histórico sobre o resultado das interações que teve com as entidades candidatas e esta base servirá de ajuda na tomada de decisão para as futuras interações.
3	Antes de qualquer interação a entidade consultora requisita opiniões de outras entidades sobre quem já possui alguma confiança. As opiniões recebidas irão determinar se entidade consultora irá interagir.
4	Antes de qualquer interação a entidade consultora requisita as opiniões das AGDs dos domínios de qual faz parte.
5	A entidade consultora combina as informações contidas em sua base de reputação própria com as opiniões recebidas de provedores de opiniões e das AGDs.

Tabela 4.2: Cenários utilizados nas simulações

Conjunto	Descrição
1	Entidades que nunca interagiram com qualquer entidade que tenha vínculo direto com a entidade consultora.
2	Entidades que já interagiram com alguma entidade que possui vínculo direto com a entidade consultora.

Tabela 4.3: Conjuntos de entidades utilizados nas simulações

vínculos de confiança entre estas entidades representam os arcos. A topologia dessas redes de confiança apresenta uma forte influência na efetividade das buscas por opiniões, uma vez que os arcos entre nós indicam como as opiniões serão propagadas.

De acordo com a literatura [Capkun et al., 2002], a topologia destas redes de confiança segue o conceito do mundo pequeno [Milgram, 1967], com uma distribuição de acordo com a lei da potência [Albert e Barabási, 2002]. Na simulações aqui realizadas optou-se por usar o modelo proposto em [Albert e Barabási, 2002] para gerar a topologia da rede e assim prover no ambiente simulado uma topologia próxima daquela que seria encontrada na prática. No capítulo 5 é apresentado um estudo aprofundado sobre a topologia das redes de confiança.

A simulação foi conduzida em uma rede com 300 entidades, sendo que 5% deste total são AGDs e as demais entidades são obrigatoriamente membros de dois domínios quaisquer. Cada entidade membro possui confiança estabelecida com 16 outras entidades, não necessariamente presentes nos mesmos domínios que a entidade em questão. Para cada cenário apresentado acima foram realizadas simulações com dois conjuntos de entidades candidatas, apresentados na tabela 4.3.

Para cada cenário foram realizados 128 experimentos, independentes entre si mas iguais em todos os cenários. Com o intuito de verificar como seria o desempenho do modelo proposto em um ambiente próximo a um cenário real em produção, ou seja, um cenário onde as entidades já detivessem uma base de experiência formada, em nossas simulações todas

entidades da rede tiveram suas bases de reputações alimentadas através de interações entre si. A entidade consultora realizou interações com cada uma das entidades com quem possui vínculo direto e o total de interações com cada entidade foi determinado de forma aleatória dentro do intervalo de 1 a 128. Para as demais entidades o total de interações realizadas com outras entidades foi determinado de forma aleatória dentro do intervalo de 1 a 512. O total de entidades que cada uma dessas interagiu também foi escolhido aleatoriamente de forma que no máximo cada entidade poderia interagir com até 1/10 do total de entidades.

4.4.1 Resultados

As figuras 4.3 e 4.4 mostram o resultado das simulações, tendo no eixo X todos os cenários sobre os dois conjuntos de entidades candidatas. As colunas representam o total de interações dividido entre os casos que resultaram em sucesso e em insucesso. A figura 4.3 apresenta um histograma sendo que no eixo *y* está expresso a quantidade real de interações realizadas em cada cenário, permitindo assim verificar quais cenários propiciaram um número maior de interações, além de indicar quantas dessas resultaram em sucesso ou em insucesso. A figura 4.4 apresenta um histograma normalizado, adequando todos os cenários dentro da faixa de 0 a 1 no eixo *y*. Assim, é possível visualizar quais cenários apresentaram uma taxa percentual de sucessos maior.

O cenário 1 apresenta o caso sem qualquer modelo de reputação sendo que a entidade consultora interagiu todas as vezes com todas entidades candidatas. Em ambos conjuntos de entidades candidatas o total de interações realizadas para cenário 1 foi de 131.072 sendo que 50% das interações terminaram em sucesso e 50% em insucesso. O cenário serve como medida comparativa, determinando o total de interações possíveis diante do conjunto de entidades candidatas.

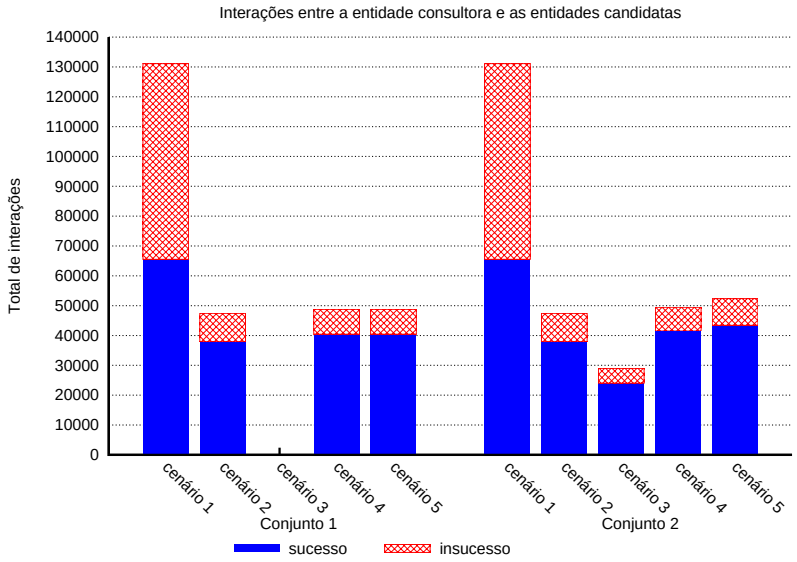


Figura 4.3: Resultado das interações entre a entidade consultora e as entidades candidatas

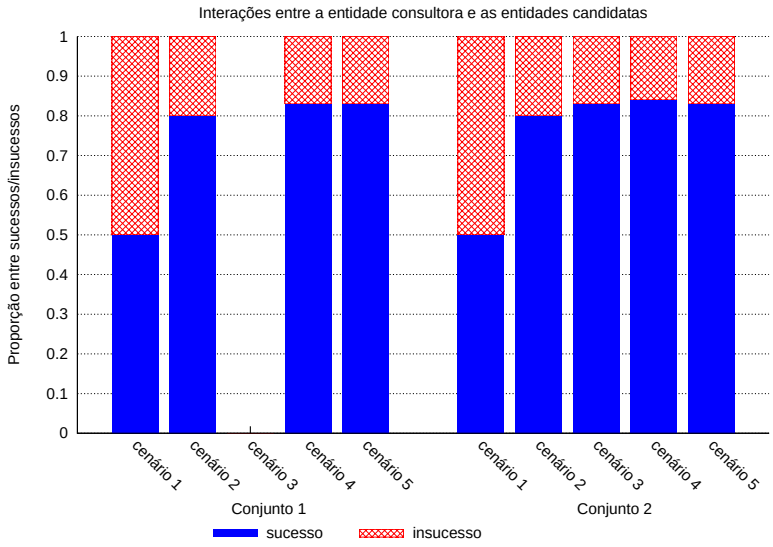


Figura 4.4: Resultado normalizado das interações entre a entidade consultora e as entidades candidatas

No cenário 2 a entidade consultora só contou com sua própria base de experiência e esta permitiu obter 80% de sucesso de um total de 47.386 interações para o conjunto 1 e 80% de sucesso de um total de 47.898 interações para o conjunto 2. No cenário 3 a entidade consultora só iria interagir se recebesse das entidades, com quem possuía vínculo, opiniões superiores ao seu limiar (veja seção 4.4). Como o conjunto 1 só continha entidades candidatas que nunca interagiram com qualquer entidade que tenha vínculo direto com a entidade consultora, nenhuma opinião foi provida e como conseqüência não foi realizada qualquer interação. Para o conjunto 2 o cenário 3 apresentou 83% de sucesso para um total de 28.928 interações.

No cenário 4 só foram solicitadas opiniões as AGDs sendo que no conjunto 1 foram realizadas 48.712 interações, com 83% de sucesso e no conjunto 2 foram 49.434 interações, resultando em 83% de sucesso. O cenário 5 combinou as bases de reputações dos cenários 2, 3 e 4 e obteve sucesso em 83% das interações em ambos conjuntos, sendo que no conjunto 1 foram realizadas 48.712 e no conjunto 2 foram 52.328. A tabela 4.4 sumariza estes números.

Com os dados da tabela 4.4 é possível concluir que o uso de sistemas de reputações

	Conjunto 1			Conjunto 2		
	Total de interações	Total de sucessos	Taxa de sucesso	Total de interações	Total de sucessos	Taxa de sucesso
Cenário 1	131.072	65.536	50%	131.072	65.536	50%
Cenário 2	47.386	37.908	80%	47.898	37.908	80%
Cenário 3	–	–	–	28.928	24.011	83%
Cenário 4	48.712	40.431	83%	49.434	41.131	83%
Cenário 5	48.712	40.431	83%	52.328	43.433	83%

Tabela 4.4: Resultados do sistema de reputações

permitiu, em média, um aumento de 30% das interações que resultaram em sucesso se comparado ao cenário 1. Apesar dos demais cenários apresentarem uma taxa de sucesso bem próximas, foi possível notar uma diferença no total de interações realizadas em cada cenário. Nos experimentos com o conjunto de entidades 1, pode-se notar que o uso das opiniões providas pela AGD apresentou melhores resultados quando comparado com os cenários que só fizeram uso de experiências diretas ou mesmo com o uso das opiniões recebidas somente de entidades com quem se tinha relações de confiança estabelecidas. Para o conjunto de entidades 2, é possível observar que apesar do cenário 5 apresentar uma taxa de sucesso semelhante aos cenários 3 e 4, teve-se um número maior de interações realizadas. No cenário 5 foram realizadas 43.433 interações que resultaram em sucesso, contra 24.011 e 41.131 dos cenários 3 e 4, respectivamente.

Com isso é possível concluir que a combinação de diferentes bases de reputações continuará resultando em uma taxa de sucesso elevada, mesmo diante de um maior número de interações. Nos cenários com bases isoladas o número de interações foi menor devido ao fato do sistema de reputação acreditar que as interações puderiam resultar em insucesso, desaconselhando assim a realização dessas interações.

4.5 Trabalhos relacionados

A literatura mostrou que muitos dos sistemas de confiança estão relacionados de certa forma com algum tipo de modelo de reputação e o modelo de confiança aqui proposto segue esta mesma linha. Diante disto, surgiu a necessidade de conhecer como os sistemas de reputações são tratados na literatura de forma que pudesse servir de inspiração para a concepção de um sistema de reputação próprio. Nesta seção são apresentados os conceitos introduzidos por alguns trabalhos e é feito um pequeno comparativo com o sistema de reputação proposto.

Em [Gray et al., 2003] é apresentada uma arquitetura de segurança que visa otimizar a formação e a propagação da confiança fazendo uso dos conceitos do *mundo pequeno* [Milgram, 1967], no qual é afirmado que cada entidade em um sistema de larga escala pode estar separada de qualquer outra entidade por somente algumas entidades intermediárias. O modelo proposto por Gray et al. [2003] tem como foco aplicações colaborativas em redes móveis *ad hoc* e baseia-se nas noções humanas sobre confiança, risco e reconhecimento. Segundo os autores, em ambientes de computação onipresente, esquemas tradicionais de autenticação, como Infra-estrutura de Chave Pública (ICP) e Kerberos [Kohl e Neuman, 1993], não são escaláveis e medidas como o "reconhecimento" [Seigneur et al., 2002] seriam mais adequadas.

A arquitetura apresentada por Gray et al. [2003] é formada pelos componentes: *entidade de reconhecimento* – ao entrar em contato com alguma outra entidade, verifica se esta é

conhecida; *análise de risco* – responsável por verificar o risco envolvido em uma possível colaboração com a entidade conhecida previamente; *gerenciamento da confiança* – gerencia as experiências anteriores realizadas com as entidades conhecidas; *controle de admissão baseado na confiança* – verifica se existe confiança suficiente para superar o risco envolvido em negociar com a entidade em questão. Assim, uma vez que a confiança é estabelecida, esta é utilizada nas decisões dinâmicas de controle de acesso. Gray et al. [2003] apresenta a equação 4.5 para realizar o cálculo do valor da confiança de uma entidade p_0 sobre uma entidade p_m.

$$Tp_0(p_m) = \frac{\sum_{k=1}^{m} (Tp_{k-1}(p_k))w_k}{m} \tag{4.5}$$

sendo $Tp_0(p_m)$ o valor de confiança que p_0 irá formar sobre um p_m qualquer. p_m é uma entidade que está m saltos distante de p_0, ou seja, entre p_0 e p_m existem $m-1$ entidades intermediárias. k é a $k-esima$ entidade intermediária entre p_0 e p_m e w_k é o peso associado a distância entre p_0 e p_k, e quanto menor for o valor de k maior será a influência do peso w_k.

Assim, segundo a equação 4.5, o valor final da confiança que p_0 terá sobre p_m é constituído pela soma de valores parciais, sendo que para cada valor parcial é descontado um valor de acordo com a quantidade de passos que uma entidade está distante de p_0, sendo tal valor representado por w_k. Ou seja, valores de confiança que foram obtidos de entidades mais próximas de p_0 terão maior influência no cálculo da confiança do que os valores que foram obtidos das entidades mais distantes. Uma vez que $Tp_0(p_m)$ esteja calculado, cabe ao p_0 determinar se este valor vai ao encontro do seu critério de seleção. Para o caso de existirem múltiplos caminhos de confiança ligando p_0 ao p_m, o modelo assume que p_0 sempre irá escolher o caminho mais confiável para assim obter o valor de confiança para p_m mais legítimo.

O modelo apresentado por [Gray et al., 2003] não possui meios para verificar se os valores de confianças fornecidos pelas entidades intermediárias são corretos, ou seja, o modelo não apresenta meios para tratar com provedores de opiniões maliciosos. Neste caso, um sistema de reputação poderia ser adotado como uma solução para evitar que entidades maliciosas interfiram no cálculo da confiança.

Em [Wang e Vassileva, 2003] é apresentado um modelo de confiança baseado em redes bayesianas que faz uso de um sistema de reputação e tem como aplicação exemplo uma rede par a par para o compartilhamento de arquivos. Cada par na rede pode assumir dois papéis, um destinado ao provimento de arquivos, denominado "provedor", e outro para o provimento de opiniões, denominado "agente". É proposta uma solução para situações onde diferentes pares possuam diferentes opiniões sobre um mesmo par. Neste caso, estes pares não são considerados maliciosos, eles apenas possuem diferentes critérios para classificar outros pares. Os autores assumem que todos os pares da rede sempre irão emitir pareceres verdadeiros

sobre a reputação de outros pares, ou seja, os pares não irão assumir um comportamento malicioso para emitir pareceres falsos.

Em suma, a rede bayesiana visa servir de apoio para que um par, ao pesquisar por arquivos, possa escolher os melhores provedores, ou seja, provedores que anteriormente se mostraram capazes em prover arquivos e por conseqüência possuem uma maior probabilidade de continuar provendo arquivos de forma satisfatória. Porém, para os casos que um "agente" não possua qualquer experiência anterior com o "provedor" é proposto um sistema de reputação, onde agentes atuam como provedores de recomendações. Um agente ao ser questionado sobre a competência de um provedor de arquivos, irá consultar em sua rede bayesiana se existe alguma opinião formada a respeito e irá responder com o valor ali presente.

O agente pode questionar sobre a reputação de um provedor para diversos outros agentes e como conseqüência irá receber diversas respostas, as quais podem ser oriundas de agentes confiáveis, não confiáveis e até de agentes desconhecidos, ou seja, agentes com quem ele não teve qualquer interação prévia. As recomendações oriundas de agentes não confiáveis são imediatamente descartadas, já as recomendações oriundas de agentes confiáveis e de desconhecidos são combinadas de acordo com a equação 4.6:

$$r_{ij} = w_t * \frac{\sum_{l=1}^{k} tr_{il} * t_{lj}}{\sum_{l=1}^{k} tr_{il}} + w_s * \frac{\sum_{z=1}^{g} t_{zj}}{g}, w_t + w_s = 1 \tag{4.6}$$

sendo r_{ij} o total de recomendações que o *i-ésimo* agente obteve sobre o *j-ésimo* provedor de arquivos. k é o número de recomendações de agentes confiáveis e g é o número de recomendações de desconhecidos. tr_{il} é o grau de confiança que o *i-ésimo* agente possui sobre o *l-ésimo* agente confiável. t_{lj} é o grau de confiança que o *l-ésimo* agente confiável possui sobre o *j-ésimo* provedor de arquivos. t_{zj} é o grau de confiança que o *z-ésimo* agente desconhecido possui sobre o *j-ésimo* provedor de arquivos. w_t e w_s são pesos definidos por cada agente para determinar a importância das recomendações feitas por agentes confiáveis e por agentes desconhecidos, respectivamente.

Uma vez calculado o grau de confiança para um determinado provedor de arquivos, é verificado se este valor é suficiente para iniciar a interação com esse provedor, sendo que cada usuário define o valor mínimo desejado para interagir. Após cada interação o agente irá atualizar sua rede bayesiana para o provedor de arquivos em questão e também irá atualizar sua confiança nos agentes que proveram as recomendações através da técnica de aprendizado por reforço [Sutton e Barto, 1998], de acordo com a equação 4.7:

$$tr_{ij}^{n} = \alpha * tr_{ij}^{o} + (1-\alpha) * e_{\alpha} \tag{4.7}$$

sendo tr_{ij}^{n} o novo grau de confiança que o *i-ésimo* agente irá possuir sobre o *j-ésimo* provedor de opiniões após a atualização. tr_{ij}^{o} representa o grau de confiança anterior. α é a taxa de aprendizado, um número real no intervalo de $[0,1]$. e_{α} é o valor da nova evidência, o qual pode ser -1 ou 1. Se o valor recomendado for maior que o necessário para iniciar a interação com o provedor de arquivo e se a interação ocorrer de forma satisfatória, então e_{α} é igual a 1 e caso a interação ocorra de forma insatisfatória, e_{α} é igual a -1.

Como visto, o trabalho [Wang e Vassileva, 2003] utiliza redes bayesianas para determinar os possíveis melhores provedores de arquivos e faz uso do aprendizado por reforço para dar manutenção em sua base de reputações sobre os agentes provedores de recomendações. A equação 4.7 garante aos agentes que forneceram recomendações verdadeiras um aumento em seu grau de confiança e aqueles que apresentarem recomendações falsas serão punidos gradativamente até atingir um limiar para serem considerados como não confiáveis. O modelo não apresenta meios para tratar casos em que agentes e provedores de arquivos atuem em conluio para que um agente correto tenha seu grau de confiança decrementado diversas vezes até atingir o ponto de ser classificado como não confiável, ou seja, o modelo apresentado por Wang e Vassileva [2003] requer uma taxa de aprendizado relativamente grande para evitar que a base de confiança seja fortemente influenciada por somente algumas poucas opiniões.

No trabalho [Carbo et al., 2003] é proposto um modelo de reputação para um sistema de agentes, denominado AFRAS, que faz uso de conjuntos difusos e considera o uso de experiências diretas e de recomendações de terceiros, agregando-as através de uma média ponderada. O modelo agrega valores de reputação novos e velhos, atribuindo pesos diferentes para cada, seguindo uma variável global denominada "memória", sendo esta > 0 e < 1. Segundo os autores, essa "memória" tem o objetivo de demonstrar como as decepções com um agente em particular afetam as atitudes futuras com outros agentes. Assim, se o grau de satisfação da última interação for similar a reputação atribuída ao agente em questão, então o grau de relevância das experiências passadas (memória) é incrementado. Caso contrário, se os valores da satisfação da última interação e a reputação do agente forem diferentes, então é incrementada a relevância da última interação.

O REGRET [Sabater e Sierra, 2001] é um modelo de confiança e reputação que considera que as recomendações podem ser providas por três diferentes tipos de origem: experiências diretas, recomendações de terceiros e através de estruturas sociais, sendo esta última o principal diferencial do REGRET em relação aos demais trabalhos na literatura. O sistema de reputação do REGRET segue uma abordagem subjetiva, como visto nos trabalhos [Wang e Vassileva, 2003; Gray et al., 2003; Jøsang, 2001], e o cálculo da reputação sobre um agente ainda considera o fator *tempo*, que concede uma maior relevância às experiências recentes, para o cálculo da confiança. Segundo os autores, tal escolha está embasada em diversos estudos psicológicos [Karlins e Abelson, 1970].

O REGRET ainda analisa a confiabilidade das recomendações recebidas sobre um de-

terminado agente, verificando a variação existente entre todas as recomendações recebidas para assim determinar o grau de confiabilidade destas. Por exemplo, uma variação pequena permite garantir que o agente em questão teve um comportamento parecido com todos os outros agentes com quem interagiu e que agora estão expressando recomendações sobre ele. Por outro lado, uma variação muito grande faz com que se tenha uma baixa confiabilidade sobre a reputação do agente.

Os trabalhos [Buchegger e Boudec, 2003; Whitby et al., 2005; Teacy et al., 2006] usam de maneira semelhante a *distribuição beta* para determinar a probabilidade de uma entidade vir a honrar a negociação. Este tipo de abordagem apresenta-se mais interessante que aquelas que fazem uso de pesos, pois possuem uma fundamentação estatística. Estes últimos trabalhos diferem entre si na forma como tratam a detecção de opiniões não confiáveis. Em [Whitby et al., 2005] as opiniões de provedores que se desviarem da maioria são descartadas. Teacy et al. [2006] compara o histórico de opiniões fornecidas por um provedor com o que de fato foi observado nas interações com a entidade sobre quem ele recebeu opiniões. Nesta última abordagem, se o provedor de opiniões fornecer de forma continuada opiniões parecidas, assume-se que o provedor é preciso, caso contrário, assume-se que as opiniões deste provedor são imprecisas e assim são descartadas.

A proposta introduzida nesta tese também faz uso de média ponderada, porém os pesos utilizados são obtidos através de métodos estatísticos, com base nas experiências observadas pela entidade que está solicitando as opiniões. Como em [Teacy et al., 2006] as opiniões recebidas consideram o histórico do provedor de opiniões, permitindo assim que as opiniões de bons provedores prevaleçam sobre as opiniões de provedores maliciosos.

Em [Teacy et al., 2006] para calcular o valor de esperado (veja equação 4.3), os provedores de opiniões devem fornecer o total de interações que resultaram em sucesso e insucesso, com a entidade objeto do cálculo da confiança. Essa abordagem fere a privacidade dos provedores de opiniões e está factível a ataque de descoberta de dados, o que permite a uma entidade maliciosa observar a evolução do provedor de opiniões com as demais entidades da rede e assim utilizar dessas informações para benefício próprio. Em nossa proposta, os provedores de opiniões só fornecem o valor esperado de forma que não se possa inferir sobre a quantidade total de casos com sucesso e insucesso.

O uso das bases de conhecimentos nas AGDs evita também uma das preocupações apresentadas por Teacy et al. [2006], que provedores de opiniões poderiam fornecer diferentes opiniões dependendo de quem as está requisitando. As bases de conhecimento dispostas na AGD só permitem incrementar os casos que resultam em sucesso e insucesso, mesmo sabendo que as entidades poderiam registrar opiniões diferentes do que realmente foi observado nas interações. Assim, uma vez que um provedor de opiniões registrou um caso de sucesso ou insucesso, essa informação estará disponível para qualquer requisitante e não poderá ser modificada.

4.6 Conclusões do capítulo

No ambiente dos Serviços *Web*, aplicações interagem com aplicações sem que necessite de intervenções por parte dos usuários. Trata-se de um ambiente dinâmico onde serviços de diferentes provedores podem ser combinados somente para atender uma determinada oportunidade de negócio e depois tal composição é desfeita.

Diante de um grande de número de provedores de serviços a escolha de um serviço diante de muitos outros, de forma automática e sem a intervenção de usuários, é um desafio que vem aos poucos despertando interesse da comunidade na proposição de soluções. Neste capítulo foi apresentado um modelo de confiança em conjunto com um sistema de reputações que visa o estabelecimento e manutenção de relações de confiança entre as entidades presentes na Arquitetura Orientada a Serviço (AOS).

Aos disponibilizarmos as bases de reputações de cada entidade em suas AGDS, evitou-se a principal preocupação apresentada pelos trabalhos na literatura. O sistema de reputação proposto apesar de permitir que uma entidade registre opiniões de forma maliciosa, este garante que as opiniões fornecidas serão concisas não importando quem as requisita, isto é, a base de reputações não pode ser manipulada de forma que apresente respostas diferentes para entidades distintas.

O sistema de reputação probabilista aqui apresentado leva em consideração as experiências passadas para assim obter o valor da confiança sobre uma determinada entidade, sendo esta provedora de serviços ou provedora de opiniões. Assim, entidades maliciosas (provedores de opiniões) serão facilmente identificados tendo suas opiniões ignoradas para o cálculo da confiança sobre as demais entidades do sistema distribuído.

Cada entidade no sistema possui uma base de conhecimento individual a qual agrupa experiências passadas e opiniões de entidades espalhadas por diferentes domínios. Tal base é então usada para que se possa formar o valor da confiança sobre uma entidade desconhecida qualquer e a interação com essa entidade só ocorre se o valor obtido atingir o limiar mínimo. Este tipo de abordagem permite ao modelo proposto lidar com o problema do **conluio**, haja visto que a individualidade das bases de conhecimento sugere que tal ataque só terá êxito se este conseguir de certa forma cobrir a ampla maioria das entidades provedoras de opiniões, tarefa a qual não é tão simples em sistemas de larga escala.

O modelo proposto nesta tese também não está suscetível ao ataque denominado **traidor**. Em tal ataque, uma entidade poderia agir corretamente diversas vezes seguidas para situações onde o seu retorno fosse pequeno e então agir de forma maliciosa em uma situação onde seu retorno fosse maior. Por exemplo, um provedor de opiniões é consultado diversas vezes em algumas negociações de venda de CDs. Se este agisse de forma maliciosa nessas negociações, teria um retorno muito pequeno. Porém, este mesmo provedor de opiniões ao ser consultado pela primeira vez sobre uma negociação relacionada a venda de um au-

tomóvel, o alto valor envolvido nesta negociação poderia compensar seu comportamento malicioso. Neste trabalho o problema com o **traidor** pode ser minimizado através dos contextos de opiniões. O provedor do exemplo anterior teria uma ótima reputação no contexto de venda de CDs, entretanto no contexto de venda de carros sua reputação seria incerta. Isto faria com que a entidade requerente consultasse as outras bases de reputações que possui.

Com os resultados obtidos nas simulações foi possível visualizar que apesar das diferentes bases de reputações apresentarem uma taxa de sucesso semelhante, a combinação de diversas bases ou mesmo o uso da base provida AGD, a qual reúne informações de todos os seus membros, permitiu que um número maior de interações fosse realizado. Conseguiu-se assim interagir com entidades que não atingiram o limiar mínimo, quando foi feito uso de somente uma base de reputação isolada, mas que o fizeram quando houve a combinação de mais de uma base de reputações.

Capítulo 5

Localização de caminhos de confiança

As relações de confiança entre as AGDs além de permitirem que membros destes domínios usufruam do conceito de autenticação única, também possibilitam que novas relações de confiança entre AGDs sejam estabelecidas. No modelo apresentado nesta tese, o estabelecimento de novas relações de confiança está condicionado a existência de um caminho de confiança entre as AGDs. Neste capítulo é apresentada a avaliação sobre os algoritmos de buscas que podem ser empregados em tal tarefa, além da proposição de um algoritmo de busca.

5.1 Introdução

A realização de qualquer interação comercial está fortemente relacionada ao nível de confiança existente entre as partes envolvidas. O estabelecimento da confiança na vida real consiste geralmente de um processo complexo e subjetivo e no comércio eletrônico tal processo apresenta-se como um desafio ainda maior [Patil e Shyamasundar, 2005]. A ausência de interações humanas combinada com a velocidade e a freqüência com as quais novas relações de negócio são estabelecidas contribuem para este desafio.

O termo confiança pode assumir diversos sentidos em uma aplicação distribuída. Em segurança o mais usual é garantir que as informações foram geradas por uma origem confiável. No caso, a preocupação geralmente restringe-se em garantir as propriedades de autenticidade e integridade sobre as informações. Diversas soluções para o estabelecimento automático da confiança foram propostas na literatura e muitas destas são amplamente utilizadas, como X.509 [Housley et al., 2002], PGP [Zimmerman, 1994] e SPKI/SDSI [Ellison et al., 1999; Rivest e Lampson, 1996]. Tais soluções apresentam meios para garantir que as informações estão sendo trocadas com uma fonte confiável.

O conceito de redes de confiança formam um caso interessante nas soluções para o estabelecimento da confiança. Nessas redes cada entidade tem habilidade para determinar com

quais outras entidades irá estabelecer confiança. O estabelecimento e o gerenciamento de relações de confiança é algo que vem sendo amplamente discutido pela literatura [Jøsang et al., 2005b; Skogsrud et al., 2003; Spantzel et al., 2005; Winslett et al., 2002]. Modelos de segurança baseados no conceito das redes de confiança eliminam o ponto único de falhas ou mesmo gargalos, tornando-os ideais para ambientes de larga escala. Os padrões PGP e SPKI/SDSI discutem ainda o conceito de *caminhos de confiança*, os quais interligam duas entidades quaisquer na rede, através de um caminho intermediado por uma ou mais entidades.

A principal dificuldade que recai sobre tais modelos consiste justamente na localização dos caminhos de confiança entre duas entidades quaisquer. Apesar dos padrões [Zimmerman, 1994; Ellison et al., 1999] descreverem os caminhos de confiança, estes não apresentam meios para realizar a busca por tais caminhos. Diversos autores propuseram algoritmos para a localização de caminhos de confiança [Atif, 2002; Santin et al., 2003; de Mello et al., 2005] visando cobrir a brecha deixada pelos padrões. Contudo, tais trabalhos não apresentam resultados em ambientes reais ou simulados com o intuito de verificar a eficiência e o desempenho desses algoritmos.

Para que se tenha um valor prático, algoritmos de busca por caminhos de confiança devem considerar que cada entidade só está ciente das demais entidades com quem possui relações diretas. Isso imediatamente sugere que algoritmos de busca em redes par a par (*Peer-to-Peer* – P2P) não estruturadas possam também ser aplicados neste domínio, como apresentado pelos trabalhos [Atif, 2002; Santin et al., 2003; de Mello et al., 2005], haja visto que nas redes P2P cada nó mantém um índice parcial que representa um subconjunto de todos os nós da rede.

Existe uma importante diferença entre as aplicações que fazem uso de redes P2P e as redes de confiança. Em uma aplicação P2P típica, como o compartilhamento de arquivos, os recursos (arquivos) estão replicados por diversos nós. Nas redes de confiança, por sua vez, as relações de confiança (recursos) só estão presentes nos dois nós que compõem este relacionamento. Apesar de ser possível replicar essa informação por diversos nós (veja seção 5.2), essa será menos expressiva do que aquela apresentada pelas aplicações P2P tradicionais.

Diversos trabalhos na literatura [Lv et al., 2002; Zhuang et al., 2003; Gkantsidis et al., 2004] apresentam simulações com algoritmos de busca em redes P2P, entretanto tais trabalhos têm como foco a busca por arquivos disponibilizados pelos nós, o que difere significativamente da busca por relações de confiança entre os nós. Este capítulo apresenta simulações realizadas com diversos algoritmos de busca destinados as redes P2P tradicionais para a localização de caminhos de confiança. Os resultados obtidos com as simulações serviram de base para a concepção do algoritmo *DiffTrust*, apresentado na seção 5.3.

5.2 Algoritmos para busca de caminhos de confiança

Em termos abstratos, as redes de confiança podem ser vistas como um grafo, sendo os nós representados pelas entidades participantes e os arcos pelas relações de confiança. Um arco de um nó *A* para um nó *B* denota que a entidade *A* possui relação de confiança com a entidade *B*. Analogamente, no PGP [Zimmerman, 1994], os nós seriam as chaves e os arcos representariam as assinaturas que quantificam a confiança [Penning, 2006]. Assim, a descoberta de caminhos de confiança equivale a busca por caminhos em grafos.

A relação de confiança entre duas entidades podem ocorrer em um único sentido ou em ambos os sentidos. Por exemplo, no modelo X.509 [Housley et al., 2002] os usuários confiam na Autoridade Certificadora (AC) mas o inverso não é verdade. Nos modelos PGP e SPKI/SDSI cada entidade pode ser o emissor ou o sujeito de uma relação de confiança, permitindo assim o estabelecimento mútuo da confiança. O modelo de confiança apresentado nesta tese segue o conceito das redes de confiança e assim sendo é assumido a existência de relações mútuas de confiança entre as entidades, o que resulta em um grafo não-direcionado.

5.2.1 Relações entre redes de confiança e redes par a par

Existem duas categorias de redes P2P descentralizadas: *não-estruturadas*, como a rede Gnutella [Gnutella, 2001]; e *estruturadas*, como as tabelas de resumos distribuídos (*Distributed Hash Table* – DHT) [Rowstron e Druschel, 2001; Stoica et al., 2003]:

Redes P2P puras e não-estruturadas. Cada nó está conectado a um subconjunto de nós da rede e a busca por recursos se faz através da inundação de mensagens pela rede. Na proposta inicial do Gnutella [Gnutella, 2001] a busca por recursos é encaminhada para um número fixo de vizinhos (geralmente para quatro vizinhos) e estes propagam a busca para seus vizinhos até que seja encontrado o recurso desejado ou até que o número máximo de saltos definido para a mensagem seja atingido (*Time To Live* – TTL);

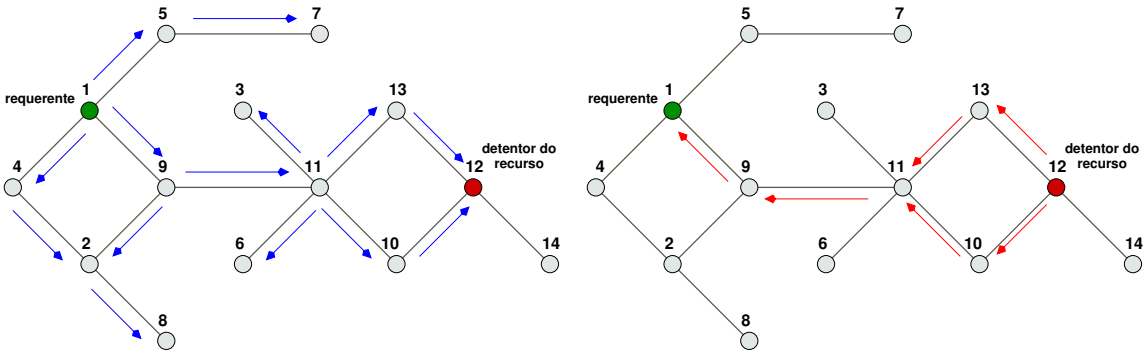
Redes P2P estruturadas. Apresentam mecanismos que controlam a topologia da rede e distribuem os recursos por nós específicos, proporcionando uma maior eficiência nas consultas subseqüentes [Lua et al., 2005]. Segundo Stoica et al. [2003] os algoritmos para as redes P2P estruturadas tendem a apresentar um melhor desempenho que os algoritmos para as redes P2P não-estruturadas pois propagam uma quantidade menor de mensagens pela rede.

Comparando as redes de confiança às redes P2P é possível constatar que estas compartilham algumas características o que permite deduzir que os algoritmos de busca, empregados nas redes P2P, possam também ser usados para encontrar caminhos em uma rede de

confiança. Contudo, existem algumas particularidades entre as redes P2P não-estruturadas e estruturadas que tornam diferentes as formas de mapeá-las para uma rede de confiança.

Em uma rede P2P não-estruturada cada nó está interconectado a um subconjunto de nós que compõem toda a rede. Cada nó disponibiliza um conjunto de recursos, por exemplo, arquivos que gostaria de compartilhar com outros nós. Nas redes de confiança as relações de confiança que um nó possui podem ser comparadas aos recursos das redes P2P.

Nas redes P2P não-estruturadas a busca por recursos é normalmente conduzida através da inundação de mensagens pela rede, isto é, o nó que está querendo encontrar algum recurso lança a busca para todos os demais nós com quem possua uma ligação direta. Esses nós, por sua vez, se forem detentores do recurso desejado, enviam uma resposta ao nó que os questionou, senão encaminham a busca para os outros nós com quem compartilham arcos. A resposta nestas redes, irá obrigatoriamente fazer um caminho inverso por onde chegou a consulta ao nó detentor do recurso. A figura 5.1(a) ilustra os caminhos percorridos pela busca disparada pelo nó 1, requisitando um recurso que só está presente no nó 12. Cada arco do grafo representa um canal de comunicação direto entre os respectivos nós. Na figura 5.1(b) tem-se o caminho percorrido pela resposta, saindo do nó 12 e chegando a origem da busca (nó 1). Com a figura 5.1 é possível notar que a resposta segue o caminho inverso àquele utilizado pelas mensagens de busca.



(a) Caminhos percorridos pela busca (b) Caminho percorrido pela resposta

Figura 5.1: Caminhos percorridos por uma busca na rede P2P não-estruturada quando empregado um algoritmo de busca por inundação

A busca por caminhos de confiança, consiste em localizar um caminho que interconecte dois nós quaisquer, podendo este caminho ser composto por nós intermediários. Por exemplo, no cenário apresentado pela figura 5.1, o nó 1 deseja encontrar um caminho de confiança que o interligue até o nó 14. Como visto nas figuras 5.1(a) e 5.1(b), o nó detentor do recurso, no exemplo o nó 12, irá responder para o nó que o imediatamente consultou e este fará o mesmo até que a resposta chegue ao nó 1. Supondo que na resposta gerada por cada nó, estes anexem sua identificação, fica fácil para o nó 1 determinar por onde a mensagem passou até chegar ao detentor do recurso, logo, tem-se conhecimento do caminho de confiança que interliga o nó 1 ao nó 14.

Algoritmos para redes P2P estruturadas geralmente são baseados em tabelas de resumos distribuídos DHT e o Chord [Stoica et al., 2003] é um protocolo exemplo dessas redes. Seu funcionamento consiste em gerar identificadores únicos (chaves) para nós e para recursos, os quais são obtidos através da aplicação de um algoritmo para geração de resumos (*hash*) sobre o endereço IP, para os nós, e sobre o conteúdo dos próprios recursos.

As chaves dos nós e dos recursos são então usadas para indicar em qual nó ficará disponibilizada a informação a respeito da localização de um determinado recurso. Assim, é possível que a informação sobre a localização de um recurso R, identificado pela chave k_{A1}, seja disponibilizada pelo nó identificado pela chave K_A, mesmo sabendo que este nó não é o verdadeiro detentor do recurso, porém é deste a responsabilidade de informar qual nó o detém. Cada nó possui ainda uma tabela de roteamento, denominada *finger table*, o que permite a um nó determinar a faixa de chaves de recursos que estarão disponíveis nos próximos $O(logN)$ nós vizinhos, sendo N o número total de nós da rede. Isto objetiva acelerar as buscas pela rede.

Como visto acima, algoritmos para redes P2P estruturadas, como o Chord, não apresentam meios para traçar um caminho entre o nó que originou a consulta e o detentor do recurso, uma vez que os algoritmos para tais redes P2P buscam criar *atalhos* entre nós justamente para reduzir a quantidade de nós visitados por uma consulta até que se encontre um recurso. Ou seja, se empregado nas redes de confiança, um nó diria que sabe quem é detentor do recurso, mesmo que ele não possua uma relação de confiança direta com o nó detentor.

Dentro do contexto deste trabalho, o Chord não apresenta meios para traçar um caminho de confiança entre o nó que originou a consulta até o nó detentor do recurso. Dessa forma, apresentamos aqui uma proposta para conseguir obter tal caminho através da realização de sucessivas buscas na rede. A figura 5.2 ilustra os passos necessários para localizar um caminho de confiança na rede apresentada pela figura 5.1, ou seja, uma rede composta por 14 nós, sendo o nó 1 o responsável pela busca e o nó 12 aquele que detém o recurso desejado.

A chave para cada nó foi gerada com a aplicação da função *hash* sobre seus identificadores, numerados de 1 a 14, resultando nos rótulos apresentados sobre as bordas do círculo. Por exemplo, *hash(nó 1) = 8*, *hash(nó 2) = 14*, ..., *hash(nó 14) = 96*. Como o algoritmo está sendo aplicado em uma rede de confiança, os recursos que cada nó compartilha são na verdade os identificadores dos nós com quem este possui relações de confiança. Dessa forma, os recursos compartilhados pelo *nó 1* são $(4,5,9)$, pelo *nó 2* são $(4,8,9)$, etc, (veja grafo da figura 5.1).

Dentro da estrutura de círculo definida pelo Chord [Stoica et al., 2003], cada recurso é identificado pela tupla *(chave,dado)*. Em nossa proposta, a *chave* representa o nó provedor do recurso e o *dado* representa a lista de nós com que este possui relações de confiança. Por exemplo, a chave para os recursos providos pelo *nó 1* é gerado através da função *hash((4,5,9))*, resultando no valor 63 (veja as tabelas apresentadas na figura 5.2).

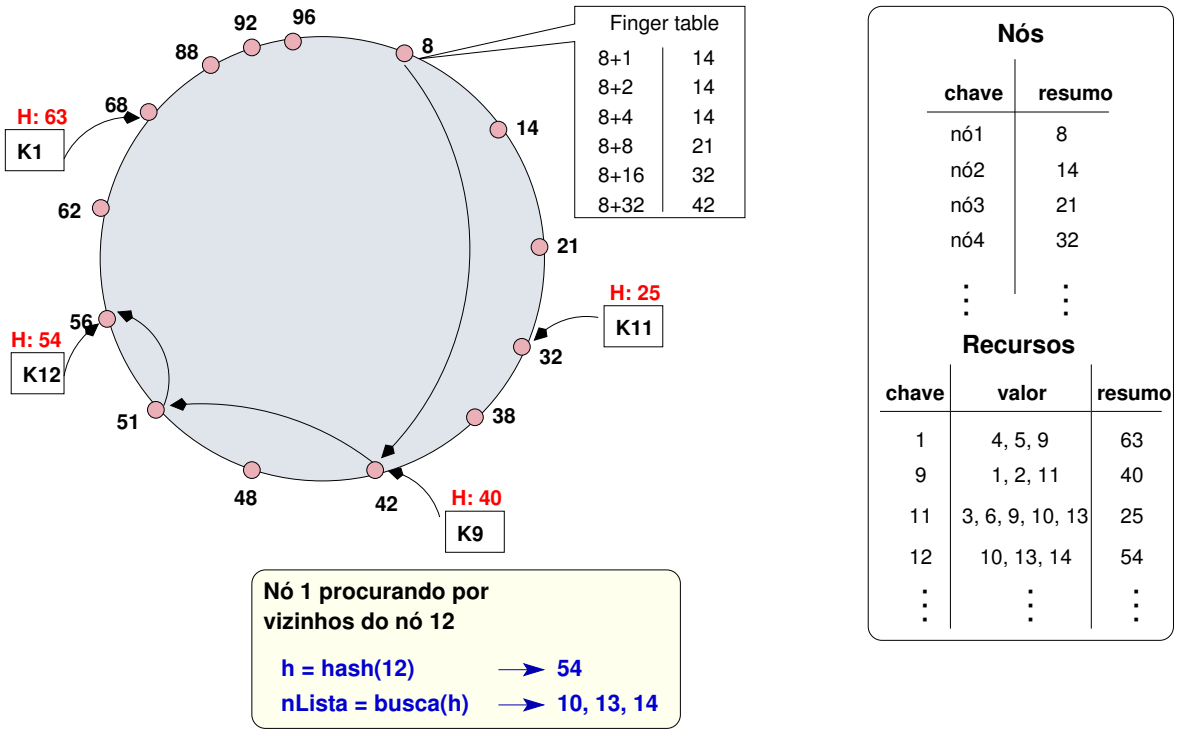


Figura 5.2: Encontrando caminhos de confiança com o Chord

Baseado nesta forma de relacionar os recursos com os identificadores dos nós, propomos o algoritmo 5.1 para realizar a busca por caminhos de confiança. Considere o exemplo no qual o *nó 1* (origem) deseja encontrar um caminho de confiança que o interconecte ao *nó 12* (alvo). A idéia básica do algoritmo consiste em montar o caminho reverso do nó alvo até chegar na origem. Busca-se inicialmente pelos nós vizinhos[1] do nó alvo e verifica se algum destes também é vizinho do nó origem. Se sim, então tem-se um caminho de confiança, intermediado por um nó, interligando o nó 1 ao 12. Caso não haja vizinhos em comum, então o nó origem lança novas buscas, porém agora procurando pelas relações de confiança dos nós contidos na lista de vizinhos do nó 12. Essa busca irá ocorrer sucessivamente até que seja encontrado um nó que possua relações de confiança com o nó 1 ou com um de seus vizinhos, caracterizando assim a existência de um caminho de confiança.

No algoritmo 5.1, no conjunto S são armazenados os identificadores dos nós que foram retornados por uma consulta. No exemplo apresentado acima, quando o *nó 1* consultou sobre os vizinhos do *nó 12*, a lista com os nós (10,13,14) foi incluída no conjunto S. Caso um dos elementos de S também esteja na lista de vizinhos do *nó 1*, a busca se encerra. Caso contrário, busca-se pelos vizinhos de cada elemento contido em S até encontrar um caminho interligando o nó 1 ao nó 12 ou até que o conjunto S torne-se vazio (linha 6). Inicialmente o conjunto S só contém o nó 12 (alvo). No conjunto R estão contidos todos os vizinhos do nó 1 (inicial). O conjunto V conterá todos os nós já questionados e que servirá de base para encontrar o caminho de confiança.

[1] Nós com quem o alvo possui relações de confiança.

Algoritmo 5.1 Busca de caminhos de confiança com o Chord

```
1:  R ← { Vizinhos do nó inicial }
2:  S ← ∅ // Conjunto de nós que serão questionados
3:  V ← ∅ // Conjunto de nós já questionados. Servirá de base para achar o caminho de confiança.
4:  found ← false
5:  S ← {alvo}
6:  while (found ≠ true OR S ≠ ∅) do
7:     q ← obtemElemento(S)
8:     V ← V ∪ {q}
9:     A ← chordQuery(q)
10:    F ← A ∩ (R ∪ {inicial})
11:    if F ≠ ∅ then
12:       found ← true
13:    else
14:       S ← S ∪ A
15:       S ← S \ V
16:    end if
17: end while
```

Na linha 7 remove-se um elemento (q) do conjunto S para o qual deseja-se conhecer a lista de vizinhos. De acordo com o exemplo anterior, inicialmente em q tem-se o nó 12. O elemento q é então adicionado ao conjunto V (linha 8), evitando que este seja consultado mais de uma vez, caso q venha a ser incluído novamente em S por alguma outra consulta subsequente. Na linha 9 é feita uma consulta para obter a lista de vizinhos do elemento q, sendo esta lista armazenada no conjunto A. Para determinar se já foi encontrado um caminho entre o nó origem o nó alvo, faz-se uma interseção dos conjuntos A e R (linha 10) e se esta não resultar em um conjunto vazio, tem-se um caminho (linha 12). Senão, é feita uma inclusão de todos elementos de A em S (linha 14), excluindo aqueles que já foram visitados (linha 15).

Na figura 5.2, as setas dentro do círculo do Chord indicam as mensagens propagadas pelo *nó 1* para encontrar os recursos providos pelo *nó 12*, no caso, sua lista de vizinhos. Como pode ser visto, foram necessárias três mensagens para que fosse possível obter a lista com os valores (10,13,14). Como os nós presentes nesta lista não são vizinhos do *nó 1*, escolhe-se um elemento desta lista (conjunto S) para buscar pelos recursos providos por este. Por exemplo, é feita uma nova busca agora procurando pela lista de vizinhos do *nó 10*, resultando nos valores (11,12). Como esses não são vizinhos do *nó 1*, são incluídos no conjunto S para que possam ser questionados posteriormente. Supondo que no próximo passo busca-se pelos vizinhos do *nó 11*, o resultado busca seria então a lista (3,6,9,10,13). Neste ponto, a lista retornada possui o *nó 9* que também é vizinho do *nó 1*, o que indica que foi encontrado um caminho que ligue o *nó 1* até o *nó 12*.

Como visto, a proposta apresentada aqui permitiu encontrar caminhos de confiança usando o Chord, ao custo de uma maior complexidade na implementação do algoritmo de busca. Trata-se de um estudo inicial e sabemos que o algoritmo pode ser otimizado, por exemplo,

através do uso de tabela *cache*, o que permitiria reduzir o número de mensagens. Contudo, ficou evidente que o uso de algoritmos para redes P2P estruturadas não apresentam benefícios diante dos algoritmos para redes não-estruturadas, pois os mesmos não foram projetados de forma que permitam montar um caminho por onde percorreu uma consulta e sua respectiva resposta. Sendo assim, optou-se por conduzir os experimentos somente com algoritmos não-estruturados.

A forma tradicional de busca empregada para os algoritmos não-estruturados consiste na inundação da rede que apesar de resultar em um grande número de respostas o faz através de um grande custo associado número de mensagens propagadas. Para a localização de caminhos de confiança esse problema é ampliado pelo fato de não haver uma grande replicação dos recursos (relações de confiança) por diversos nós na rede.

Diversos trabalhos propuseram melhorias ao método de inundação, como a busca em profundidade [Gkantsidis et al., 2004; Lv et al., 2002; Chawathe et al., 2003] e a busca por largura que consiste no aumento gradativo do TTL [Zhuang et al., 2003; Jiang e Jin, 2005; Yang e Garcia-Molina, 2002]. A rede Kazaa [Kazaa] introduziu o conceito de super-nós, também chamados de *ultra-peers*, criando uma hierarquia na rede. Nesta rede as consultas são propagadas somente na camada dos super-nós, e as ligações entre esses super-nós atuam como atalhos entre os nós da camada inferior. A seção seguinte apresenta detalhes sobre o cenário utilizado para a realização dos experimentos com estas abordagens.

5.2.2 Experimentos com algoritmos de inundação

Como visto, diversos trabalhos propuseram medidas para aprimorar o desempenho dos algoritmos de inundação com o intuito de diminuir o número de mensagens propagadas na rede. Em [Gkantsidis et al., 2004; Lv et al., 2002; Zhuang et al., 2003; Chawathe et al., 2003; Jiang e Jin, 2005] são apresentadas avaliações sobre tais abordagens dentro do contexto de aplicações destinadas ao compartilhamento de arquivo. Entretanto, a literatura não apresenta avaliações sobre o comportamento dessas abordagens na localização de caminhos de confiança. Dessa forma, nesta seção é feito um comparativo entre estas abordagens através de experimentos em um ambiente simulado. As abordagens implementadas foram as seguintes:

K-caminhos. Cada nó propaga a consulta para k vizinhos selecionados de forma aleatória, o que resulta em uma variação da busca por largura;

Consulta seletiva. Semelhante a abordagem dos *k-caminhos*, porém a consulta só será propagada para os k melhores nós de acordo com um critério específico, por exemplo, localização, largura de banda, número de vizinhos que possui, resultados obtidos com as consultas anteriores, etc. Na busca por caminhos de confiança fica claro que o ideal

é selecionar os nós que possuem mais vizinhos, o que implica que estes possuem mais relações de confiança;

Busca em largura. Essa abordagem consiste em incrementar repetidamente o valor do *Time To Live* (TTL) associado as consultas até que um caminho de confiança seja encontrado. Inicialmente a busca com um valor baixo para o TTL (expresso em termos de número de saltos) e caso não seja encontrado o recurso, então o valor do TTL é incrementado e a mesma busca é reencaminhada. Esse processo se repete até que o valor máximo para o TTL seja atingido. Essa abordagem consegue evitar o problema quando um recurso já foi encontrado nas proximidades mas o TTL ainda não atingiu seu limite, o que resulta na propagação de mensagens desnecessárias pela rede;

Super-nós. Essa abordagem introduz uma hierarquia entre os nós. Quando um nó folha ingressa na rede, este associa-se a um ou mais super-nós. No contexto das redes de confiança, cada super-nó armazena informações sobre as relações de confiança de seus nós folhas. A busca iniciada por um nó folha será propagada somente na camada dos super-nós, uma vez que esses nós conhecem as relações de confiança que seus nós folhas possuem;

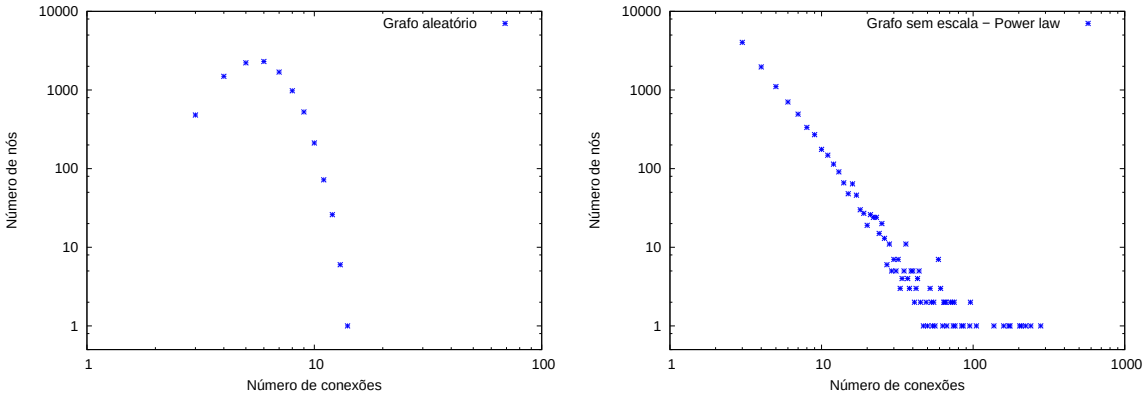
Tabela *cache*. Nesta abordagem cada nó da rede possui uma tabela *cache* que armazena referências sobre as consultas anteriores que resultaram em sucesso. Se for realizada uma nova busca tendo o mesmo alvo, a resposta será obtida mais rapidamente. Essa abordagem pode ser combinada com qualquer um dos algoritmos apresentados acima e nas simulações optou-se por combiná-la somente com a proposta tradicional de inundação, como a do Gnutella [Gnutella, 2001].

5.2.2.1 Topologias

A topologia de uma rede P2P influencia fortemente a eficiência dos algoritmos de busca. A topologia é determinada pelo número de vizinhos que um nó possui (o grau de um nó) e neste estudo optou-se por estudar duas classes de topologias: grafo aleatório e grafo baseado no modelo *sem escala* (*scale free*) [Albert e Barabási, 2002].

Existem diferentes variações de grafos aleatórios e diversas abordagens para gerar tais grafos. Nas simulações aqui realizadas seguiu-se a abordagem apresentada pelo simulador Peersim [Peersim], a qual gera para cada nó um número fixo d de arcos e os conecta a d nós selecionados de forma aleatória. Uma vez que as simulações são conduzidas sobre grafos não-direcionados, tal abordagem resulta em um grau médio de $2d$ para cada nó. A figura 5.3(a) ilustra (usando uma escala logarítmica) o número de nós com um certo grau, sendo 4 o grau médio para um nó.

Grafos *sem escala* seguem a distribuição baseada na lei de potência (do inglês, *power law*) em que muitos nós possuem poucas conexões e poucos nós possuem muitas conexões,



(a) Distribuição de conexões em um grafo aleatório (b) Distribuição de conexões em um grafo sem escala

Figura 5.3: Distribuição de conexões em diferentes topologias de rede

como pode ser observado na figura 5.3(b). Este tipo de distribuição representa o conceito do *mundo pequeno* [Milgram, 1967], o qual é observado em diferentes áreas, inclusive nas redes de confiança [Capkun et al., 2002] e em aplicações P2P tradicionais para o compartilhamento de arquivos [Gnumap]. O simulador Peersim apresenta também uma implementação para a geração de grafos *sem escala*, a qual segue a abordagem apresentada por Albert e Barabási [2002].

As simulações foram conduzidas através do simulador de eventos discretos Peersim [Peersim], o qual permite a criação de redes de acordo com diversas topologias, incluindo grafos aleatórios e grafos sem escala. O simulador provê meios para a criação de réplicas independentes dos experimentos, o que permite que a mesma topologia seja usada em simulações com diferentes algoritmos de busca.

5.2.2.2 Resultados

Existem diversas métricas que podem ser usadas para qualificar os algoritmos de busca por caminhos de confiança. Para alguns casos, basta encontrar um único caminho de confiança para sanar as necessidades da aplicação. Em outros casos, a obtenção de múltiplos caminhos de confiança podem trazer uma maior garantia para a aplicação. Em ambos os casos, o custo do algoritmo pode ser medido através do número de mensagens propagadas pela rede.

Nesta seção são apresentados resultados somente para o primeiro caso, ou seja, só é desejado conhecer o custo associado a cada algoritmo para encontrar um único caminho de confiança, tendo ciência que os algoritmos estudados podem resultar em mais caminhos. Em suma, a busca encerra-se assim que um caminho de confiança for encontrado. É também analisada a sensibilidade dos resultados com relação ao valor do *Time To Live* (TTL), uma vez que a ausência da noção de tempo em nossas simulações pode ser contornada através do uso do TTL como um indicador de tempo que um algoritmo leva para encontrar caminhos de confiança. Os resultados são ilustrados pela figura 5.4 e pelas tabelas 5.1 e 5.2.

As simulações foram realizadas em uma rede composta por 20.000 nós para ambas topologias, grafo aleatório e grafo sem escala, sendo 4 o grau médio do nó. Foram feitas simulações com um nó alvo em três diferentes distâncias a partir de um nó origem (escolhido de forma arbitrária): um nó mais próximo – a 2 saltos da origem; um nó a uma distância média; e o nó mais distante do nó origem. Para a topologia do grafo aleatório o nó a uma distância média ficou a 7 saltos e o mais distante ficou a 10 saltos da origem. Para a topologia do grafo sem escala o nó a uma distância média ficou a 4 saltos e o mais distante ficou a 6 saltos. Para todos os algoritmos, e em ambas topologias, o valor do TTL começou em 2 e foi sendo incrementado até 7, sendo algumas vezes incrementado além disso para observar um fenômeno específico. Para os algoritmos *consulta seletiva* e *k-caminhos* foram escolhidos três diferentes valores para o número de vizinhos que a busca será propagada: 10%, 50% e 70%. No algoritmo *super-nós* foram selecionados, de forma aleatória, os nós mais conectados para assumirem o papel de super-nós. A quantidade de nós promovidos a super-nós se deu através da parte inteira de $\sqrt{N}$, sendo N o tamanho da rede.

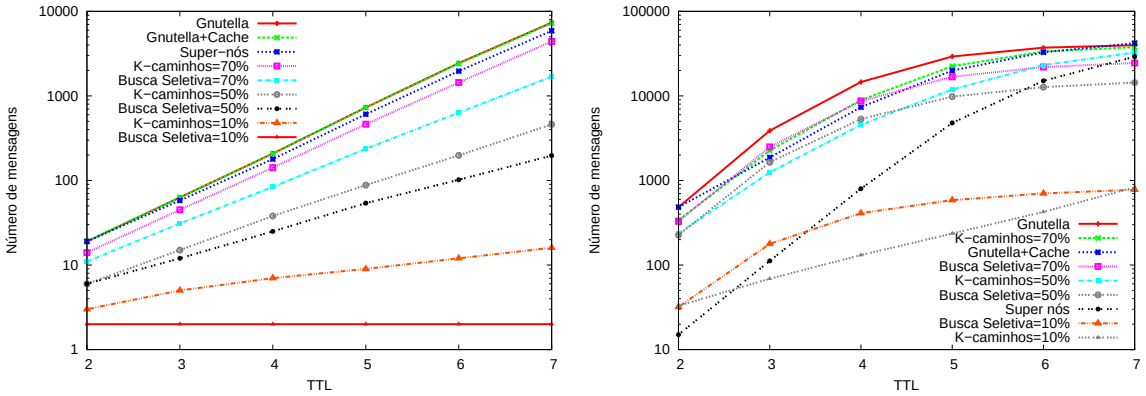
TTL	Gnutella		K-caminhos			Busca seletiva			Super-nós
	original	cache	10%	50%	70%	10%	50%	70%	
5	0	0	0	0	0	0	0	0	0
6	2	2	0	0	2	0	0	1	2
7	3	3	0	0	2	0	0	1	3
10							1º		
11				1º					
32			1º						

Tabela 5.1: Número de caminhos de confiança encontrado para a topologia de grafo aleatório

TTL	Gnutella		K-caminhos			Busca seletiva			Super-nós
	original	cache	10%	50%	70%	10%	50%	70%	
2	0	0	0	0	0	0	0	0	1
3	1	2	0	1	1	1	1	1	3
4	4	19	0	3	2	1	2	3	7
5	4	95	0	3	3	1	2	3	9
6	5	138	1	4	4	1	3	3	10
7	5	180	1	5	5	1	3	3	11

Tabela 5.2: Número de caminhos de confiança encontrado para a topologia de grafo sem escala

Buscou-se apresentar os resultados para o caso tendo o nó alvo a uma distância média, assim foi escolhido um nó a 7 saltos do nó origem no grafo aleatório e a 4 saltos no grafo sem escala. As tabelas 5.1 e 5.2 apresentam a quantidade de caminhos de confiança encontrados, para diferentes valores do TTL, para as topologias de grafo aleatório e grafo sem escala, respectivamente. As últimas três linhas da tabela 5.1 indicam o valor do TTL necessário para encontrar pelo menos um caminho de confiança, porém mesmo com grandes valores para o TTL o algoritmo "busca seletiva – 10%" não conseguiu encontrar um caminho.



(a) Grafo aleatório – nó alvo a 7 saltos

(b) Grafo sem escala – nó alvo a 4 saltos

Figura 5.4: Número de mensagens sob diferentes valores para o TTL

Os gráficos apresentados nas figuras 5.4(a) e 5.4(b) ilustram o número de mensagens propagadas pela rede para cada algoritmo, sob diferentes valores do TTL. É possível constatar que tanto a topologia da rede quanto o algoritmo adotado, apresentam uma forte influência no número de mensagens propagadas. Na rede com a topologia *sem escala* alguns nós possuem muitos vizinhos o que resulta em uma maior quantidade de mensagens propagadas pela rede quando algoritmos de inundação são usados. Entretanto, tais mensagens são redundantes pois uma mesma consulta é propagada para um mesmo nó mais de uma vez. Em [Lv et al., 2002] é apresentada uma discussão mais aprofundada sobre tal fenômeno.

Apesar de resultar em um número maior de mensagens propagadas pela rede, a topologia *sem escala* apresentou o benefício de encontrar um caminho de confiança com apenas poucos saltos, como pode ser observado nas tabelas 5.1 e 5.2. Em todos os cenários foi possível encontrar um caminho de confiança com o valor do TTL = 3, exceto para o cenário K-caminhos 10%. No grafo aleatório um caminho de confiança nem sempre pode ser encontrado com valores baixos para o TTL e assim algoritmos que não propagam a consulta para um grande número de vizinhos requerem valores altos para o TTL. A tabela 5.1 apresenta estes valores nas três últimas linhas, sendo que para encontrar um único caminho de confiança os algoritmos k-caminhos 50% e k-caminhos 10% requerem um $TTL = 11$ e $TTL = 32$, respectivamente. O algoritmo busca seletiva 10% em nenhum dos casos encontrou um caminho.

Para os grafos *sem escala* os algoritmos *busca seletiva* e *k-caminhos*, percorrendo somente 10% dos vizinhos foram os que apresentaram melhores resultados, propagando somente 178 e 425 mensagens pela rede, respectivamente. Para efeito comparativo, o algoritmo de inundação tradicional (Gnutella) gerou quase 4.000 mensagens até encontrar um caminho de confiança. No grafo aleatório os algoritmos *busca seletiva* e *k-caminhos* só apresentaram um desempenho melhor que o Gnutella para o caso onde a consulta é propagada por 70% dos vizinhos.

Os algoritmos *busca seletiva* e *k-caminhos* são alternativas naturais à inundação. Con-

tudo, a *busca seletiva* mostrou ser uma alternativa ainda melhor que o algoritmo *k-caminhos*, devido a dependência de fatores aleatórios por este último, tornando difícil a previsão dos resultados em termos do número de mensagens necessárias. Para encontrar um caminho de confiança com $k = 50\%$ no grafo aleatório o algoritmo $k - caminhos$ apresentou o pior desempenho, necessitando de 9.000 mensagens, quatro vezes mais do que o número necessário pelo algoritmo de inundação tradicional. Nas simulações apresentadas nesta seção o algoritmo de *busca seletiva* propagou a consulta somente para os nós que possuem mais conexões.

Para o algoritmo de inundação combinado à tabela *cache* foram realizadas diversas buscas consecutivas usando um mesmo nó origem e um mesmo nó destino. Os valores obtidos com a primeira consulta foram usados para iniciar as tabelas *cache*. Na segunda vez que a consulta foi gerada, todos os nós que possuíam o recurso (caminho de confiança) em sua tabela *cache* respondiam a consulta. Essa mesma consulta repetiu-se até que todos os vizinhos do nó origem tivessem valores em suas tabelas. Dentro do nosso ambiente de testes foram necessárias quatro consultas consecutivas para que isso fosse atingido, o que resultou nos dados apresentados pelas tabelas 5.1, 5.2 e figura 5.4.

O algoritmo *super-nós* na topologia de grafo aleatório despendeu uma quantidade de mensagens muito semelhante ao algoritmo de inundação tradicional. Contudo, no grafo sem escala o algoritmo *super-nós* obteve melhores resultados, encontrando caminhos de confiança com poucas mensagens. A razão para isto se deve ao fato da promoção dos nós com um grande número de conexões a super-nós, fato o qual funciona bem nas redes com a topologia sem escala e nem tanto para as redes com topologia aleatória.

Em todos os experimentos o valor do TTL começou baixo e foi incrementado gradativamente até atingir o limite estipulado. Este comportamento se assemelha ao algoritmo de *busca em largura*, dessa forma os possíveis resultados com tal abordagem podem ser obtidos através da soma dos resultados individuais para cada valor do TTL, do algoritmo Gnutella.

5.3 O algoritmo de busca *DiffTrust*

Os experimentos apresentados na seção anterior serviram para delimitar a abordagem mais adequada para a localização dos caminhos de confiança. É possível concluir que a abordagem da busca seletiva aliada à abordagem da busca em largura seria uma boa opção, como apresentado em [Zhuang et al., 2003]. Assim, uma solução mista surgiu de motivação para a proposição do *DiffTrust*. Na seção 5.3.1 é feita uma pequena discussão indicando como é feita a classificação das relações de confiança, a qual é usada pelo algoritmo *DiffTrust*. Por fim, a seção 5.3.2 apresenta o funcionamento do algoritmo *DiffTrust*.

5.3.1 Classificação das relações de confiança

O algoritmo *DiffTrust* segue o conceito da inundação, porém apresenta um comportamento diferenciado de acordo com a classificação das relações de confiança. Cada Autoridade de Gerência do Domínio (AGD) na rede de confiança classifica suas relações como sendo *relações fortes* ou *relações fracas*. Tal classificação está fundamentada no modelo de confiança apresentado no capítulo 4 sendo que para cada relação de confiança é associado um peso. Os pesos são números reais delimitados dentro do intervalo de 0 a 1, sendo 0 o valor para representar a confiança mais baixa e 1 para representar a confiança mais alta. Cabe a cada AGD definir seu próprio limiar (h) para determinar quais relações serão tratadas como fortes ou fracas. Por exemplo, relações que tiverem um peso acima $0,5$ serão consideradas relações fortes, ou fracas caso contrário.

Cada domínio possui uma *base de conhecimento* própria, composta pelo histórico das interações realizadas por seus membros (veja seção 4.2). Essa base permite a cada AGD conhecer os resultados das interações que seus membros tiveram com membros de outros domínios, que por conseqüência, são regidos por outras AGDs. Tal base é então usada pela AGD para determinar os pesos que estarão associados a cada relação com outras AGDs.

Como apresentado na seção 4.2.1, um membro m para calcular a reputação sobre uma entidade t, recorre ao valor esperado ($E(m,t)$) (equação 4.3) de uma distribuição beta, o qual indica o quanto é confiável a entidade t do ponto de vista do membro m. Seguindo essa mesma idéia, uma AGD_f calcula a reputação para uma outra AGD_t, combinando o valor esperado de todos os seus membros. Assim, o valor esperado para a AGD_t, calculado pela AGD_f é determinado pela equação 5.1:

$$R_f(t) = E(beta(\sum_{m \in M} \alpha_{m,t}, \sum_{m \in M} \beta_{m,t})) \tag{5.1}$$

sendo que $R_f(t)$ representa a reputação da AGD_t e M o conjunto de membros da AGD_f.

Cada AGD define seu próprio limiar h para determinar se uma relação de confiança será *forte* ou *fraca*. Desta forma, o relacionamento entre a AGD_f e a AGD_t será forte somente se $R_f(t) > h_f$, caso contrário será considerada uma *relação fraca* (h_f representa o limiar da AGD_f).

Os membros de cada domínio são os responsáveis por alimentar a bases de conhecimento das AGDs, contudo cada membro possui uma visão particular na rede e é possível que diferentes membros tenham diferentes percepções sobre uma mesma interação. Por exemplo, os clientes c_1 e c_2 interagiram com um mesmo provedor de serviço p_1. Ambos obtiveram acesso ao recurso desejado x minutos depois de o terem requisitado. Para o cliente c_1 a interação ocorreu com sucesso e assim registra a satisfação em sua base de conhecimento. Considerando que o cliente c_2 precisava obter o recurso $x - y$ minutos depois do pedido e

isto não ocorreu, c_2 registra em sua base que a interação ocorreu com insucesso. Tem-se então opiniões divergentes o que resulta em dificuldades para um sistema de reputações. Os clientes podem não ser maliciosos, mas o fato de possuírem diferentes pontos de vista podem tornam duvidosa a reputação de um provedor de serviço correto.

Para tratar tal problema, optamos pela abordagem apresentada por Whitby et al. [2005], a qual descarta opiniões que divergem da maioria. Assim, somente as opiniões que representam a maioria dos membros de um domínio são consideradas no cálculo do valor esperado $R_f(t)$ de uma determinada AGD. Essa abordagem foi escolhida diante de muitas outras, pois as AGDs não são entidades ativas, ou seja, uma AGD não tem como verificar se uma opinião recebida está correta pois esta não interage, por exemplo, com provedores de serviço para assim ter uma observação própria. Assumimos assim, que cada AGD possui um papel passivo no modelo.

Cada AGD apresenta seus próprios limites (inferior e superior) para descartar opiniões que assumirem valores extremos. Esses limites são representados por p, sendo o limite inferior representado por p e o limite superior representado por $(1-p)$. Opiniões que estiverem fora dessa faixa são descartadas. Deve-se escolher um valor para p que seja suficientemente baixo para evitar que se exclua muitas opiniões corretas e significativamente alto para evitar a inclusão de opiniões incorretas. Dessa forma, opiniões de membros que resultarem no valor esperado $((1-p) \times R_f(t)) < E_{m,t} < (p \times R(t))$ são descartadas.

5.3.2 Funcionamento do algoritmo *DiffTrust*

As *relações fortes* possibilitam a uma AGD delegar a busca por um caminho de confiança a outras AGDs, cabendo a estas últimas a responsabilidade em consultar as demais AGDs com quem possuam relações de confiança. Em suma, relações fortes proporcionam um comportamento idêntico àquele apresentado por algoritmos tradicionais de inundação. Se uma AGD, ao ser consultada sobre um recurso e não possuir o mesmo, esta propagará a consulta para todas as AGDs com quem possui relações de confiança. As *relações fracas* não possibilitam tal delegação e uma AGD ao receber uma consulta, e se não possuir o recurso desejado, irá responder para a AGD que a consultou com a lista das AGDs com quem esta possui relações de confiança. Assim, cabe a AGD que originou a busca, consultar as demais AGDs presentes nesta lista.

A figura 5.5 ilustra o caminho percorrido por uma busca através de uma rede composta por relações fortes e fracas. Cada nó do grafo representa uma AGD, as linhas tracejadas representam as relações de confiança fracas entre AGDs, as linhas contínuas representam as relações de confiança fortes e as setas indicam as mensagens propagadas pela rede.

No exemplo apresentado pela figura 5.5, a AGD “A” deseja encontrar um caminho de confiança que a conecte à AGD “J” e assim propaga uma busca por todos os seus vizinhos

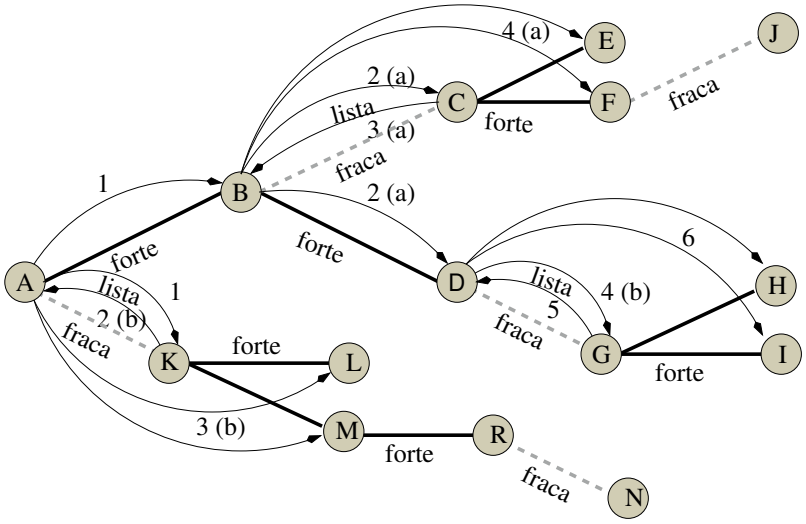


Figura 5.5: Algoritmo *DiffTrust*

(passo 1). Como existe uma relação forte entre "A" e "B", a AGD "B" se encarrega de propagar a busca por seus vizinhos (passo 2a). Entre "A" e "K" existe uma relação fraca e assim "K" retorna para "A" a lista com todas as AGDs com quem "K" possui relações de confiança (passo 2b). Neste ponto cabe a "A" consultar diretamente todas as AGDs presentes na lista fornecida por "K", no caso, "L" e "M" (passo 3b). Tal comportamento também pode ser observado entre "B" e "C" assim como entre "D" e "G". Por fim, ao questionar "F", a AGD "B" obtém uma resposta positiva a qual é encaminhada pelo caminho inverso percorrido pela busca até atingir a AGD que a originou, neste caso a AGD "A".

As *relações fracas* também atuam como mecanismo para limitar a profundidade da busca. Quando uma busca passa por um caminho composto por uma relação fraca, esta busca só poderá ser propagada por mais um nível no grafo. Isto é, a busca só será encaminhada para as AGDs contidas na lista fornecida por uma AGD com quem se tem uma relação fraca. Por exemplo, existe uma relação fraca entre as AGDs "A" e "K". "A" encaminha uma busca a "K" e esta por sua vez responde com uma lista contendo as AGDs com quem possui relações de confiança (passo 2b). "A" consulta as AGDs presentes na lista (passo 3b) e a busca encerra-se, uma vez que as AGDs nesta lista ("L" e "M") não propagarão a consulta e também não possuem o recurso desejado. Por outro lado, caminhos compostos por *relações fortes* subseqüentes, o limite de profundidade da busca é definido por um tempo de vida que diminui a cada salto (*Time To Live* – TTL), o qual pode ser definido de acordo com os requisitos de cada aplicação.

A principal idéia por trás dos diferentes níveis de confiança (forte e fraca) consiste em limitar o número de mensagens propagadas pela rede para a busca de um caminho de confiança sem que isso venha a diminuir o desempenho da abordagem de inundação. Dessa forma, as relações fortes continuam permitindo um comportamento semelhante a inundação tradicional enquanto que as relações fracas ajudam a limitar o número de mensagens propagadas pela rede.

Algoritmo 5.2 Comportamento de um nó ao receber uma consulta

Require: $V = \{$ Conjunto de todos os nós com quem possui relações de confiança $\}$
1: **if** $(recurso \subset V)$ **then**
2: $responde$(origem, $\varnothing$, verdade) // Respondendo ao nó origem que possui o recurso
3: **else if** $(ttl > 0)$ **then**
4: $N \leftarrow V \setminus \{origem\}$ // Remove o elemento origem do conjunto V e armazena em N
5: **if** (relação forte com $origem$) **then**
6: **while** $N \neq \varnothing$ **do**
7: $x \leftarrow getElement(N)$
8: $consulta$(x, noAtual, recurso, $ttl - 1$)
9: $N \leftarrow N \setminus \{x\}$ // Remove o elemento x do conjunto N
10: **end while**
11: **else if** (relação fraca com $origem$) **then**
12: $responde$(origem, N, falso) // Enviando ao nó origem a lista de vizinhos.
13: **end if**
14: **end if**

O algoritmo 5.2 apresenta a computação realizada por um nó ao receber uma consulta. Primeiramente o nó verifica se possui o recurso desejado (linha 1). Caso o possua, então responde ao nó que o consultou (linha 2), invocando a função `reponde`, cujo primeiro parâmetro indica a qual nó enviará a resposta (`nóOrigem`), o segundo parâmetro contém um conjunto vazio (`listaVizinhos`) e no terceiro é um campo booleano o qual indica que o recurso foi encontrado (`verdadeiro`). Se o nó consultado não possuir o recurso, então dois diferentes comportamentos podem ser assumidos, de acordo com o peso associado a relação de confiança entre este nó e aquele que o consultou. Se a relação for forte, então a consulta é encaminhada para todos os seus vizinhos (linhas 6–10); caso contrário (relação fraca) o nó retorna a sua lista de vizinhos àquele nó que lhe consultou (linha 12).

O comportamento de um nó ao receber uma resposta (enviada pelas linhas 2 ou 12 do algoritmo 5.2) é apresentado no algoritmo 5.3. Um nó pode ter originado a busca na rede ou pode ter somente reencaminhado a busca após recebê-la de seu vizinho. Em ambos os casos, o conjunto G representa os nós que já foram consultados por este.

Na linha 1 do algoritmo 5.3 é verificado se o recurso foi encontrado ou não, de acordo com o valor fornecido no parâmetro booleano da função *responde* do algoritmo 5.2. Se for verdade, então na linha 2 é testado se o nó em questão foi quem originou a busca na rede. Se sim, então encerra-se a busca e retorna-se à aplicação para o processamento da regra de negócio (linha 3). O conjunto N conterá então todo o caminho percorrido pela busca. Caso a condição na linha 2 seja falsa, então o nó atual invoca a função *responde*, incluindo a si próprio no conjunto N, o qual representará o caminho percorrido pela resposta. Por fim, se *achou* for falso (linha 1), então a busca é propagada para todos os elementos presentes no conjunto N, exceto para aqueles já presentes no conjunto G (linhas 9 – 16).

Algoritmo 5.3 Comportamento de um nó ao receber uma resposta

Require: $G = \{$ Conjunto de todos os nós já consultados previamente $\}$
Require: $N = \{$ Conjunto de nós retornado por uma consulta prévia $\}$
1: **if** $achou = verdade$ **then** // Indica se houve resposta positiva
2: **if** este nó foi quem originou a busca = $verdade$ **then**
3: Fim da busca e processa o conjunto N // Em N estará o caminho percorrido pela busca
4: **else**
5: $N \leftarrow N \cap \{noAtual\}$ // Conjunto que contém o caminho percorrido para encontrar o recurso
6: $responde$(origem, N, verdade)
7: **end if**
8: **else**
9: $R \leftarrow G \cap N$ // Identificando os nós que não deverão ser consultados
10: $C \leftarrow N \setminus R$ // Removendo os nós presentes em G para não serem consultados novamente
11: **while** $C \neq \varnothing$ **do**
12: $x \leftarrow getElement(C)$
13: $consulta$(x, noAtual, recurso, $ttl - 1$)
14: $G \leftarrow G \cup \{x\}$
15: $C \leftarrow C \setminus \{x\}$
16: **end while**
17: **end if**

5.3.3 Experimentos e resultados

A efetividade do algoritmo *DiffTrust* foi verificada através de simulações, cujos resultados foram comparados com um algoritmo de inundação tradicional, como o Gnutella [Gnutella, 2001]. O comparação consistiu em verificar o número de caminhos encontrados e a quantidade de mensagens propagadas pela rede. As simulações foram conduzidas de forma similar àquelas apresentadas na seção 5.2.2, isto é, foi usado o simulador Peersim [Peersim], um grafo não direcionado com a topologia *sem escala* [Albert e Barabási, 2002].

A principal diferença nestas simulações para aquelas realizadas na seção 5.2.2, foi a associação de pesos aos arcos, os quais indicam se a relação entre os nós é forte ou fraca. No modelo proposto nesta tese, a atribuição e a manutenção destes pesos se dá através do sistema de reputações apresentado no capítulo 4. Com isso, o número de relações fortes e fracas de uma rede torna-se dinâmico e em constante alteração durante todo o tempo de vida da aplicação distribuída.

Nesta seção gostaríamos de conhecer o desempenho do algoritmo *DiffTrust* em diferentes cenários, ou seja, em redes de confiança compostas por uma quantidade diferente de relações fortes. Como gostaríamos de quantidades pontuais sobre as relações fortes, optou-se por montar ambiente simplificado, cuja distribuição dos pesos associados as relações ocorreu de forma pseudo-aleatória. Optou-se por essa abordagem, pois o uso do sistema de reputação apresentado no capítulo 4 exigiria regras de negócios complexas para que conseguíssemos um número preciso de relações fortes e fracas.

Optamos em realizar as simulações em quatro diferentes cenários, os quais podem ser compreendidos como fotos dos grafos em dados momentos (momentos em que a rede dinâmica estaria congelada). A adoção destas fotos deve-se ao fato que as relações de confiança estão em constante alternância entre *fortes* e *fracas*. No primeiro cenário, denominado *DiffTrust-4k*, aproximadamente 4.000 relações foram definidas como fortes. Nos demais cenários, *DiffTrust-15k*, *DiffTrust-19k* e *DiffTrust-39k*, foram assumidas, 15.000, 20.000 e 39.996 relações fortes, respectivamente. O cenário *DiffTrust-39k* representa uma rede de confiança sem distinções entre as relações, fazendo com que o comportamento em tal cenário seja similar àquele apresentado pela abordagem de inundação tradicional.

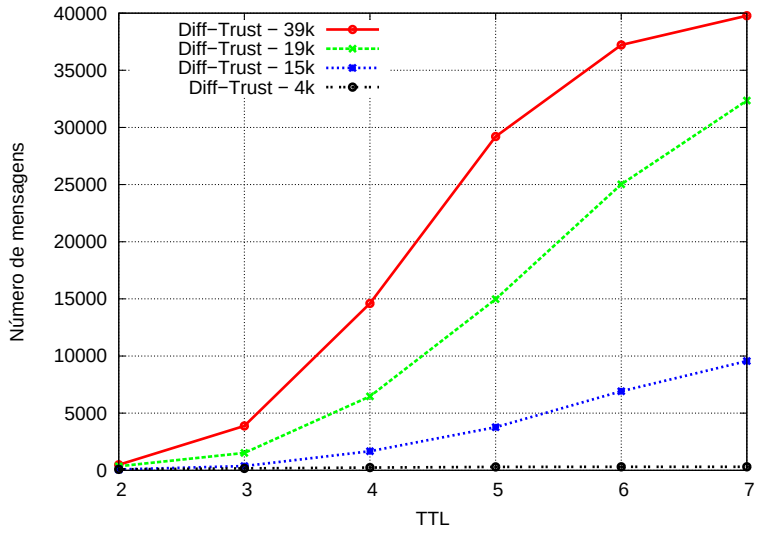


Figura 5.6: Algoritmo *DiffTrust* - número de mensagens sob diferentes TTL

O gráfico apresentado na figura 5.6 mostra o número de mensagens propagadas em cada cenário sob diferentes valores para o TTL. Fora o cenário *DiffTrust-4k*, todos os demais conseguiram apresentar respostas, isto é, conseguiram encontrar caminhos de confiança, sendo que os cenários *DiffTrust-15k* e *DiffTrust-19k* apresentaram resultados somente a partir do $TTL = 4$ e o cenário *DiffTrust-39k* apresentou resultados a partir do $TTL = 3$. De qualquer forma, a busca nesses cenários foi conduzida para valores maiores do TTL para que se pudesse constatar o comportamento dos mesmos diante de valores elevados para o TTL.

A princípio o comportamento do algoritmo *DiffTrust* parece gerar uma quantidade maior de mensagens quando comparado com a abordagem tradicional de inundação, uma vez que nós (com relações fracas) irão gerar mensagens de respostas (provendo a lista de vizinhos) mesmo quando não possuírem o recurso. No Gnutella, um nó só irá gerar respostas se este possuir o recurso ou se este estiver no caminho reverso de uma consulta que apresentou uma resposta. Entretanto, o *DiffTrust* consegue compensar essa troca extra de mensagens, limitando a profundidade das buscam que passem por caminhos compostos por relações fracas.

Considerando o número de mensagens propagadas *versus* o número de respostas obti-

das, é possível concluir que o *DiffTrust* apresenta um desempenho melhor que a abordagem tradicional de inundação. Os experimentos mostraram que é possível encontrar respostas positivas mesmo diante de poucas relações fortes. Para o pior caso do *DiffTrust*, em termos de mensagens propagadas na rede, em que todas as relações são fortes, o comportamento deste algoritmo será similar aos algoritmos de inundação, como o Gnutella.

5.3.4 Trabalhos na literatura

O fato dos principais padrões [Zimmerman, 1994; Ellison et al., 1999] negligenciarem a busca por caminhos nas redes de confiança, surgiu de motivação para proposição de diferentes soluções na literatura. Em Santin [2004] é apresentado um estudo sobre diversas soluções nesta área, além de apresentar uma forma própria para navegação nas teias de confiança através da abordagem de busca em largura. Os trabalhos [Atif, 2002; de Mello et al., 2005] propõem o uso de um algoritmo tradicional de inundação para localizar caminhos de confiança entre dois sujeitos, assumindo assim a semelhança entre as redes par a par e as redes de confiança. Entretanto, tais trabalhos não apresentam experimentos, em um ambiente real ou simulado, para verificar o desempenho das soluções propostas. Neste capítulo foram realizados experimentos sobre o algoritmo de inundação o que nos leva a pensar que os resultados obtidos seriam semelhantes aos dos trabalhos [Atif, 2002; de Mello et al., 2005].

Considerando as redes de confiança como um tipo de rede par a par, buscou-se conhecer trabalhos que se propunham a concepção de algoritmos de busca para redes par a par de forma a verificar se tais abordagens caberiam no contexto das redes de confiança. Diante disto, em [Lv et al., 2002] é proposto uma modificação no algoritmo de inundação, fazendo com que a busca percorra somente k caminhos. O foco do trabalho consiste em aplicações P2P para compartilhamento de arquivos e assim, apresenta simulações sobre diversos tipos de topologias, variando ainda a porcentagem de replicação dos recursos pelos nós da rede.

Segundo Lv et al. [2002], algoritmos de inundação não apresentam um bom desempenho em grafos com a topologia *sem escala* simplesmente por causa da presença de nós que possuem um grande número de conexões, fazendo com que o número de mensagens duplicadas seja maior do que nos grafos de topologias aleatória. Tal fato também foi observado nas simulações realizadas na seção 5.2.2. Contudo, a afirmação apresentada por Lv et al. [2002] que grafos *sem escala* não são ideais para as aplicações de compartilhamento de arquivos, não pode ser mantida para o cenário das buscas por caminhos de confiança pois a principal diferença entre as aplicações consiste no grau de replicação dos recursos pelos nós da rede. Vale ressaltar que a topologia de grafo *sem escala* se faz presente nas redes par a par ou mesmo nas redes de confiança, conforme constatado pelos trabalhos [Gnumap; Tangmunarunkit et al., 2002; Capkun et al., 2002].

Segundo Zhuang et al. [2003] o uso exclusivo de algoritmos de busca só em largura ou só

em profundidade não é o ideal, visto que o primeiro gera um grande número de mensagens e o segundo resulta em uma cobertura parcial da rede. Em [Zhuang et al., 2003] é apresentado o algoritmo híbrido de inundação periódica que busca reduzir o número de mensagens geradas, sem que isto acarrete em uma cobertura parcial da rede. O número de vizinhos pela qual a busca será propagada em largura dependerá do valor do TTL ao longo do caminho da busca. Por exemplo, quanto maior for o valor do TTL menor será a quantidade de nós visitados naquele nível. O trabalho faz uso do algoritmo *Busca em largura*, apresentado em [Yang e Garcia-Molina, 2002], para controlar a profundidade da busca. No caso, uma política determinará os valores de TTL, iniciando com um valor baixo que é incrementado a cada falha no processo de busca.

5.4 Conclusões do capítulo

Algoritmos de redes P2P são candidatos óbvios para encontrar caminhos de confiança e neste capítulo foi apresentada uma comparação entre os algoritmos de busca para redes P2P quando aplicados nas redes de confiança. Com base nos experimentos é possível concluir que os algoritmos tiveram um bom desempenho diante de redes com a topologia sem escala, especialmente quando comparados às aplicações tradicionais de compartilhamento de arquivos.

Os resultados obtidos também serviram de motivação para a proposição do algoritmo *DiffTrust*, o qual visou combinar os benefícios apresentados pelas diferentes abordagens de inundação. É importante frisar que os experimentos apresentados neste capítulo se continham na localização de um único caminho de confiança entre dois nós. Contudo, em alguns cenários específicos seria interessante obter múltiplos caminhos de confiança de forma a quantificar o grau de confiança de um determinado nó obtido diante de toda a rede. Zhang e van Moorsel [2008] apresenta uma avaliação sobre os mesmos algoritmos apresentados neste capítulo, porém verificando o desempenho destes para a obtenção de múltiplos caminhos de confiança.

Capítulo 6

Conclusões

Esta tese apresentou um modelo de segurança voltado para o ambiente dos Serviços *Web* com o intuito de permitir não apenas a troca segura de mensagens nas interações entre clientes e provedores de serviço, mas também para permitir a integração de diferentes tecnologias de segurança, empregadas por estas entidades.

O modelo apresentado segue o conceito de federações, agrupando clientes e provedores de serviços em domínios de segurança, sendo estes regidos por uma entidade denominada Autoridade de Gerência do Domínio (AGD). Trata-se de um modelo com mecanismos de autenticação centralizados e de autorização descentralizados. Assim, à AGD além das tarefas para controle de membros do domínio, tem-se ainda outras atribuições como a emissão e validação de asserções de segurança.

O estabelecimento de relações de confiança entre AGDs permitem que credenciais emitidas em um domínio possam ser reconhecidas em outros domínios, contudo ainda é necessário que as credenciais possam ser compreendidas por qualquer entidade independente da tecnologia de segurança subjacente e neste trabalho isto foi coberto pelo uso do *Security Assertion Markup Language* (SAML). A transposição de credenciais de segurança pelos diversos domínios possibilitou que os membros destes domínios usufruíssem do conceito da autenticação única (*Single Sign-On* – SSO). Nesta tese as relações de confiança além de permitirem a transposição de credenciais, também são empregadas por entidades membros, clientes e provedores de serviço, com o intuito de sinalizar com quais entidades possuem mais ou menos afinidade.

O gerenciamento das relações de confiança, de membros e de AGDs, fez com que o trabalho fosse desdobrado em duas partes. Uma primeira parte preocupou-se em definir um modelo de confiança aliado a um sistema de reputação. Tal modelo permitiu a cada entidade do sistema ponderar as relações de confiança que possui com as demais entidades. Uma outra parte ficou voltada para a localização de caminhos de confiança, esta específica ao gerenciamento da confiança entre AGDs. Por se tratar de um modelo de confiança igualitário,

sem qualquer tipo de hierarquia na confiança, tem-se a necessidade de mecanismos para a localização de caminhos de confiança que interliguem duas entidades quaisquer e como fora apresentado, tal mecanismo é negligenciado pela literatura.

6.1 Revisão dos objetivos

Nesta seção são revisados os objetivos apresentados na seção 1.2, indicando como o modelo proposto satisfaz tais objetivos.

O principal objetivo desta tese consistiu na proposição de um modelo de segurança voltado para o ambiente dos Serviços *Web* possibilitando que clientes e provedores de serviços pudessem interagir mesmo diante de diferentes tecnologias de segurança empregadas nas camadas subjacentes. Assumiu-se que a solução proposta deveria fazer uso de padrões amplamente aceitos de forma a garantir o principal atrativo dos Serviços *Web*, o qual consiste na integração de aplicações.

Para atingir a solução proposta, o objetivo geral foi dividido em objetivos específicos. Tais objetivos são apresentados a seguir, bem como uma revisão sobre como estes foram atingidos:

- **Integrar aplicações que fazem uso de diferentes tecnologias de segurança**. No modelo apresentado nesta tese, as propriedades básicas de segurança são garantidas através do uso padrões de segurança, como o XMLEnc e XMLDSign, apresentados na seção 2.2.2. A concepção dos domínios de segurança, apresentados na seção 3.2, organizou clientes e provedores de serviços de acordo com a tecnologia de segurança. A AGD assumiu um papel central a cada domínio intermediando as interações entre seus membros e o uso do SAML combinado as relações entre AGDs propiciou a transposição das credenciais de segurança, permitindo que membros de um domínio de segurança pudessem interagir com membros de outros domínios;
- **Permitir a cada entidade do modelo meios para gerenciar a confiança**. Ao agrupar clientes e provedores de serviços em domínios, assumiu-se que estes, denominados membros, expressam confiança em suas respectivas AGDs. Contudo, tais membros podem ainda expressar a confiança em outros membros, mesmo que estes estejam em diferentes domínios, como apresentado na seção 3.3. O capítulo 4 apresentou o modelo de confiança aliado a um sistema de reputação o qual provê meios para que membros possam determinar suas relações de confiança, além de uma forma para quantificá-las. Com isso foi apresentada uma solução para cenários onde o cliente, diante de muitos provedores de serviços, possa escolher um provedor que tenha sido o mais regular em suas interações passadas. No sistema de reputação apresentado, tanto membros quanto as AGDs são fontes de reputações e as informações providas por esses ajudam as demais entidades do modelo no gerenciamento da confiança;

- **Prover meios para o estabelecimento dinâmico da confiança**. No sistema de reputação apresentado no capítulo 4, membros e AGDs são fontes de opiniões os quais são consultadas por outras entidades para a manutenção das relações de confiança existentes ou para o estabelecimento de novas relações. As relações de confiança entre as AGDs possibilitaram que membros de outros domínios atuassem também como fontes de opiniões. As relações entre AGDs também serviram de base para o estabelecimento das novas relações entre AGDs, haja visto que o modelo de confiança apresentado nesta tese segue o conceito das redes de confiança. Neste tipo de abordagem, o estabelecimento de novas relações de confiança pode estar condicionado a existência de caminhos de confiança e a localização de tais caminhos é algo que sempre foi negligenciado pelos padrões. Assim, no capítulo 5 foi apresentado um estudo sobre algoritmos de buscas que poderiam ser adotados para a localização de caminhos além da proposição de um algoritmo específico para tal função.

6.2 Contribuições e resultados da tese

Esta tese teve como foco a proposição de um modelo de segurança para um ambiente dinâmico e heterogêneo e o desenvolvimento desta resultou em algumas contribuições apresentadas a seguir:

- Definição de um modelo de segurança o qual agrupa as entidades presentes na Arquitetura Orientada a Serviço (AOS) em domínios de acordo com a tecnologia de segurança subjacente. Para o gerenciamento dos membros de cada domínio e a interação entre domínios foi introduzida a Autoridade de Gerência do Domínio (AGD), permitindo assim que credenciais de segurança emitidas em um domínio pudessem ser trocadas e compreendidas em outros domínios o que possibilitou a autenticação única *Single Sign-On* (SSO) mesmo diante de diferentes tecnologias de segurança. O protótipo implementado serviu para comprovar que o modelo sugerido é factível, integrando duas tecnologias de segurança, o SPKI e o X.509. As propostas anteriores apresentaram soluções para autenticação única, porém somente entre domínios que fizessem uso de uma mesma tecnologia de segurança subjacente;

- Concepção de um modelo de confiança aliado a um sistema de reputações possibilitando a cada entidade do modelo manter uma base própria de experiências, a qual é usada nas tomadas de decisões, por exemplo, para determinar qual outra entidade interagir diante de uma nova oportunidade de negócio. A disponibilização do sistema de reputação nas AGDs aliado a uma nova abordagem para a ponderação das opiniões, se fez como uma solução para as principais preocupações levantadas pela literatura, como visto na seção 4.5. Tal modelo propiciou o estabelecimento dinâmico da confiança, algo ainda não apresentado pela literatura, dentro do ambiente dos Serviços *Web*;

- Estudo sobre a adequação de algoritmos de busca para redes par a par na localização de caminhos de confiança. O uso de simulações em uma topologia de rede bem próxima daquela encontrada em um ambiente real propiciou determinar o custo, através do número de mensagens propagadas na rede, para a localização de pelo menos um único caminho de segurança. Na literatura as simulações com tais algoritmos se continham em aplicações para o compartilhamento de arquivos;

- Proposição de um algoritmo para a localização de caminhos de confiança o qual faz uso do nível de confiança já existente entre as entidades que compõem a rede de confiança. As simulações conduzidas indicaram que o algoritmo proposto consegue encontrar caminhos de confiança a um custo menor (número de mensagens propagadas) que um algoritmo de inundação tradicional.

Alguns dos resultados obtidos com os estudos realizados nesta tese foram divulgados na forma de publicações, sendo 6 nacionais, 5 internacionais e duas submissões para revista. São estas:

- **Nacionais**

  - Ancajima, G. M. C., de Mello, E. R., e da Silva Fraga, J. "Integração da arquitetura de segurança dos Serviços *Web* com modelos de confiança igualitária". VI Simpósio Segurança em Informática (SSI'04), São José dos Campos, SP - Brasil, 2004. ITA.
  - de Mello, E. R., da Silva Fraga, J., e Camargo, E. "Transferência de autenticação e autorização através de Serviços Web". V Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSEG'05), Florianópolis, SC - Brasil, 2005.
  - de Mello, E. R., da Silva Fraga, J., e Santin, A. O. "O uso do SPKI/SDSI em redes P2P". I Workshop sobre Redes Peer-to-Peer (WP2P'05), Fortaleza, CE - Brasil, 2005. XXIII Simpósio Brasileiro de Redes de Computadores (SBRC'05).
  - de Mello, E. R., Wangham, M. S., da Silva Fraga, J., e Camargo, E. Segurança em Serviços Web, capítulo 1, pp. 1-48. Minicursos do SBSeg 2006. Sociedade Brasileira de Computação, 2006.
  - Camargo, E., da Silva Fraga, J., Wangham, M. S., e de Mello, E. R. "Autenticação e Autorização em Arquiteturas Orientadas a Serviço através de Identidades Federadas". Simpósio Brasileiro de Redes de Computadores (SBRC'07), Belém, PA - Brasil, 2007. SBRC.
  - Wangham, M. S., Boger, D., de Mello, E. R., e da Silva Fraga, J. "Implementação de um Serviço XKMS para o SPKI/SDSI com suporte as Federações SPKI". Salão de Ferramentas do SBRC 2007, Belém, PA, 2007.

- **Internacionais**
  - de Mello, E. R. e da Silva Fraga, J. "Mediation of trust across Web Services". 3rd IEEE International Conference on Web Services (ICWS'05), pp. 515-522, Orlando, Flórida - EUA, 2005.
  - de Mello, E. R., Wangham, M., da Silva Fraga, J., e Rabelo, R. "A secure model to establish trust relationships in web services for virtual organizations". 6th IFIP Working Conference on Virtual Enterprises (PRO-VE'05), Valência, Espanha, 2005.
  - de Mello, E. R., Parastatidis, S., Reinecke, P., Smith, C., van Moorsel, A., e Webber, J. "Secure and provable service support for human-intensive real-state processes". IEEE International Conference on Services Computing (SCC'06), pp. 495-502, Chicago - EUA, 2006.
  - Wangham, M. S., de Mello, E. R., da Silva Fraga, J., e Boger, D. "A Model to support SPKI Federations management through XKMS". IEEE International Conference on Web Services, 2007 (ICWS'07), Salt Lake City, Utah - EUA, 2007.
  - de Mello, E. R., van Moorsel, A., e da Silva Fraga, J. "Evaluation of P2P Search Algorithms for Discovering Trust Paths". European Performance Engineering Workshop, volume 4748, pp. 112-124. Springer-Verlag, 2007.
- **Revista indexada**
  - de Mello, E. R., Wangham, M. S., Boger, D., Camargo, E., da Silva Fraga, J. "A Model to Authentication Credentials Transposition in Service Oriented Architecture". "Security in Computing" Springer Transactions on Computational Science, 2009. Em processo de publicação.
- **Submissões para revista**
  - de Mello, E. R., Wangham, M. S., da Silva Fraga, J. "Introducing a reputation-based trust model in federated XKMS environment". Submissão a IEEE Transactions on Services Computing. Em processo de revisão.

6.3 Perspectivas futuras

A idéia inicial desta tese consistia em apresentar uma solução de segurança para o ambiente dos Serviços *Web* permitindo que clientes e provedores de serviços pudessem interagir, de uma forma fácil e automática, mesmo diante de diferentes tecnologias de segurança e mesmo que estes nunca tivessem interagido anteriormente, ou seja, permitindo assim o estabelecimento dinâmico da confiança entre partes estranhas.

A transposição de credenciais apresentada na seção 3.2.1 é um tema que pode ser explorado em trabalhos futuros em duas frentes. Esta tese restringiu-se no uso de duas tecnologias de segurança, apresentando um conjunto padrão de atributos para a transposição de credenciais SPKI em X.509 e *vice-versa*. Caberia então realizar um estudo sobre o uso de outras tecnologias como o Kerberos e credenciais biométricas.

Outro ponto não tratado por este trabalho foi a questão da privacidade das informações dos usuários. Em um cenário ideal, os usuários poderiam exercer o direito de determinar como suas informações serão manipuladas, informando quais informações poderão ser compartilhadas com terceiros, como esse compartilhamento deve ser feito e também indicando o período de tempo o qual essas informações poderão ficar disponíveis nos sistemas. O projeto *Shibboleth* [Shibboleth, 2005] apresenta uma preocupação com a privacidade das informações dos usuários, definindo como requisitos da arquitetura, meios para gerenciar quais informações um sítio origem irá transferir para um sítio destino, com o consentimento do usuário. Com o crescimento do uso de Serviços *Web* a questão da privacidade ganha um foco ainda maior, visto que um fluxo de negócios pode ser composto por diversos Serviços *Web*, ultrapassando assim diversos domínios administrativos e de segurança. A especificação W3C [2004] apresenta algumas considerações sobre a privacidade na arquitetura dos Serviços *Web*, indicando que tal assunto ainda não está completamente solucionado e necessita de um estudo mais aprofundado.

A interação entre clientes e provedores de serviços depende da combinação das políticas de segurança de ambas partes. Por exemplo, o cliente indica que só aceita interagir se as mensagens trocadas forem cifradas com uma chave de 128 *bits* e a política do provedor de serviços indica que este atua com chaves de 64 e 128, possibilitando assim a interação entre estes. A especificação WS-Policy [WS-Policy, 2004] provê um modelo de propósito geral para descrever polítcas e como associá-las a documentos WSDL e elementos da UDDI. Contudo, atualmente não existe um conjunto padrão de políticas de autorização que defina requisitos mínimos para clientes e provedores de serviços. Trata-se de uma área interessante para investigar e que a literatura começa a apresentar os primeiros trabalhos sobre a mesma [da Silva Böger et al., 2008].

Nas redes de confiança cada entidade indica em quem deseja confiar e os caminhos de confiança permitem tais entidades fazerem uso da confiança indireta para que então possam interagir. As principais especificações neste área não apresentam meios para localizar tais caminhos o que serviu de motivação para diversos autores na literatura. Devido a similaridade entre as redes de confiança e as redes par a par (P2P), nesta tese foi feito um estudo comparativo entre os algoritmos empregados nas redes P2P para localizar caminhos de confiança, contudo o trabalho se ateve em verificar o número de mensagens propagadas na rede para localizar pelo menos um caminho de confiança. Em certas aplicações quanto mais caminhos de confiança forem encontrados, entre duas entidades quaisquer, maior será o grau de confiança entre estas. Assim, determinar quais algoritmos seriam menos custosos

para encontrar mais caminhos seria outra área de estudo, a qual inclusive já está começando a ser explorada na literatura [Zhang e van Moorsel, 2008].

A integração e o funcionamento de aplicações envolvendo diferentes políticas e domínios de segurança compõem um problema complexo e de difícil solução. Esperamos que as soluções apresentadas neste trabalho, as quais se restrigiram ao gerenciamento da confiança, venham a ajudar na melhor compreensão da segurança neste ambientes complexos.

Referências Bibliográficas

Albert, R. e Barabási, A.-L. "Statistical Mechanics of Complex Networks". *Reviews of Modern Physics*, 74:47, 2002.

Apache. *Axis Architecture Guide v1.2*. The Apache Software Foundation, 2005. `http://ws.apache.org/axis/java/architecture-guide.pdf`.

Asokan, N., Schunter, M., e Waidner, M. "Optimistic protocols for fair exchange". Em *4th ACM Conference on Computer and Communications Security (CCS'97)*, pp. 7–17, New York, NY, USA, 1997. ACM Press. ISBN 0-89791-912-2.

Atif, Y. "Building Trust in E-Commerce". *IEEE Internet Computing*, 6(1):18–24, 2002.

Bartel, M., Boyer, J., e Fox, B. *XML-Signature Syntax and Processing*. W3C, fevereiro de 2002. `http://www.w3.org/TR/xmldsig-core`.

Blaze, M., Feigenbaum, J., Ioannidis, J., e Keromytis, A. D. "The role of trust management in distributed systems security". Em *Secure Internet programming: security issues for mobile and distributed objects*, pp. 185–210, London, UK, 1999. Springer-Verlag. ISBN 3-540-66130-1.

Brown, N. e Kindel, C. *Distributed Component Object Model Protocol – DCOM/1.0*. Microsoft, novembro de 1996.

Buchegger, S. e Boudec, J.-Y. L. "A Robust Reputation System for Mobile Ad-hoc Networks". Relatório Técnico IC/2003/50, EPFL IC, 2003.

Capkun, S., Buttyan, L., e Hubaux, J.-P. "Small worlds in security systems: an analysis of the PGP certificate graph". Em *New Security Paradigms Workshop*, pp. 28–35, setembro de 2002.

Carbo, J., Molina, J., e Davila, J. "Trust management through fuzzy reputation.". *International Journal of Cooperative Information Systems*, 12(1):135–155, 2003.

Chawathe, Y., Ratnasamy, S., Breslau, L., Lanham, N., e Shenker, S. "Making gnutella-like P2P systems scalable". Em *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'03)*, pp. 407–418. ACM, agosto de 2003.

Conklin, A., Dietrich, G., e Walz, D. "Password-Based Authentication: A System Perspective". Em *37th Annual Hawaii International Conference on System Sciences (HICSS'04)*, volume 7, p. 70170b. IEEE, 2004.

da Silva Böger, D., Wangham, M. S., da Silva Fraga, J., e Mafra, P. M. "Um Modelo de Composição de Políticas de Qualidade de Proteção para Serviços Web Compostos". Em *Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais (SBSeg'08)*, Gramado, RS - Brasil, 2008.

Daemen, J. e Rijmen, V. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer-Verlag, 2002.

Damiani, E., di Vimercati, S. D. C., e Samarati, P. "Managing Multiple and Dependable Identities". Em *IEEE Internet Computing*, pp. 29–37. IEEE, novembro de 2003.

de Mello, E. R., da Silva Fraga, J., e Santin, A. O. "O uso do SPKI/SDSI em redes P2P". Em *I Workshop sobre Redes Peer-to-Peer (WP2P'05)*, Fortaleza, CE - Brasil, 2005. XXIII Simpósio Brasileiro de Redes de Computadores (SBRC'05).

de Mello, E. R., van Moorsel, A., e da Silva Fraga, J. "Evaluation of P2P Search Algorithms for Discovering Trust Paths". Em *European Performance Engineering Workshop*, volume 4748, pp. 112–124. Springer-Verlag, 2007.

Dierks, T. e Allen, C. *The TLS Protocol – Version 1.0*. IETF RFC 2246, janeiro de 1999.

DoD, D. "Trusted Computer System Evaluation Criteria". DoD 5200.28-STD, dezembro de 1985.

Eastlake, D. e Jones, P. *US Secure Hash Algorithm 1 (SHA1)*. Internet Engineering Task Force RFC 3174, setembro de 2001.

Ellison, C. M., Frantz, B., Lampson, B., Rivest, R., Thomas, B. M., e Ylonen, T. *SPKI Certificate Theory*. Internet Engineering Task Force RFC 2693, setembro de 1999.

Freier, A. O., Karlton, P., e Kocher, P. C. *The SSL protocol - v.3*. Internet Draft, março de 1996.

Gambetta, D. *Trust: Making and Breaking Cooperative Relations*. Basil Blackwell, 1988.

Gkantsidis, C., Mihail, M., e Saberi, A. "Random Walks in Peer-to-Peer Networks". Em *INFOCOM*, 2004.

Gnumap. "GnuMap Project". `http://home.comcast.net/~gregory.bray`, 2002.

Gnutella. *The Gnutella Protocol Specification v0.4*. Clip2, 2001.

Grandison, T. e Sloman, M. "A Survey of Trust in Internet Applications". *IEEE Communications Surveys and Tutorials*, 3(4):2–16, 2000.

Gray, E., Seigneur, J.-M., Chen, Y., e Jensen, C. D. "Trust Propagation in Small Worlds". Em *First International Conference on Trust Management*, pp. 239–254, maio de 2003.

Hallam-Baker, P. e Mysore, S. H. *XML Key Management Specification (XKMS 2.0)*. W3C – Proposed Recommendation, maio de 2005.

Housley, R., Polk, W., Ford, W., e Solo, D. *Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF RFC 3280, abril de 2002.

IBM e Microsoft. *Security in a Web Services World: A Proposed Architecture and Roadmap*. IBM Corporation and Microsoft Corporation, abril de 2002. `http://msdn.microsoft.com/ws-security/`.

Imamura, T., Dillaway, B., e Simon, E. *XML Encryption Syntax and Processing*. W3C, dezembro de 2002. `http://www.w3.org/TR/xmlenc-core`.

InComm. "InComm Federation: Common Identity Attributes". `http://www.incommonfederation.org/docs/policies/federatedattributes.pdf`.

Internet2 e EduCause. "eduPerson". `http://www.educause.edu/eduperson`.

Jain, R. *The art of computer systems performance analysis*. Wiley, 1991.

Jiang, H. e Jin, S. "Exploiting Dynamic Querying like Flooding Techniques in Unstructured Peer-to-Peer Networks". Em *International Conference on Network Protocols (ICNP)*, pp. 122–131. IEEE Computer Society, 2005. ISBN 0-7695-2437-0.

Jøsang, A. "A Logic for Uncertain Probabilities". *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–212, 2001.

Jøsang, A., Fabre, J., Hay, B., Dalziel, J., e Pope, S. "Trust requirements in identity management". Em *Australasian workshop on Grid computing and e-research (CRPIT'44)*, pp. 99–108, Darlinghurst, Australia, 2005a. Australian Computer Society, Inc. ISBN 1-920-68226-0.

Jøsang, A., Keser, C., e Dimitrakos, T. "Can We Manage Trust?". Em *3rd International Conference on Trust Management (iTrust'05)*, pp. 93–107, 2005b.

Jøsang, A. e Pope, S. "User Centric Identity Management". Em *Asia Pacific Information Technology Security Conference (AusCERT'05)*, maio de 2005.

Karlins, M. e Abelson, H. I. *Persuasion, how opinion and atitudes are changed*. Crosby Lockwood & Son, 1970.

Kazaa. "Kazaa Media Desktop". `http://www.kazaa.com`, 2001.

Khare, R. e Rifkin, A. "Trust Management on the World Wide Web". *Computer Networks*, 30(1-7):651–653, 1998.

Kohl, J. e Neuman, C. *The Kerberos Network Authentication Service (v5)*. Internet Engineering Task Force RFC 1510, setembro de 1993.

Lampson, B., Abadi, M., Burrows, M., e Wobber, E. "Authentication in Distributed Systems: Theory and Practice". *ACM Transactions on Computer Systems*, 10(4):265–310, 1992.

Landwehr, C. E. "Computer Security". Em *International Journal of Information Security*, volume 1, pp. 3–13. Springer-Verlag Heidelberg, julho de 2001.

Liberty. *Introduction to the Liberty Alliance Identity Architecture*. Liberty Alliance, março de 2003.

Lorch, M., Proctor, S., Lepro, R., Kafura, D., e Shah, S. "First experiences using XACML for access control in distributed systems". Em *ACM Workshop on XML Security*, outubro de 2003.

Lua, E. K., Crowcroft, J., Pias, M., Sharma, R., e Lim, S. "A Survey and comparison of peer-to-peer overlay networks schemes". Em *Communications Surveys & Tutorials*, number 2, pp. 72–93. IEEE, 2005.

Lv, Q., Cao, P., Cohen, E., Li, K., e Shenker, S. "Search and replication in unstructured peer-to-peer networks". *16th International Conference on Supercomputing*, pp. 84–95, 2002.

Milgram, S. "The small world problem". *Psychology Today*, 1:61, 1967.

Neuman, B. C. *Readings in Distributed Computing Systems*, capítulo Scale in distributed systems, pp. 463–489. IEEE Computer Society, Los Alamitos, CA, 1994.

OASIS. *Universal Description, Discovery and Integration v3.0.2 (UDDI)*. Organization for the Advancement of Structured Information Standards (OASIS), outubro de 2004a.

OASIS. *Web Services Security: SOAP Message Security 1.0*. OASIS, março de 2004b. `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf`.

OASIS. *Authentication Context for the OASIS Security Assertion Markup Language (SAML) v2.0*. Organization for the Advancement of Structured Information Standards, março de 2005a.

OASIS. *eXtensible Access Control Markup Language (XACML) version 2.0*. Organization for the Advancement of Structured Information Standards (OASIS), fevereiro de 2005b. `http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf`.

OASIS. *SAML Executive Overview*. Organization for the Advancement of Structured Information Standards (OASIS), março de 2005c.

OASIS. *Security Assertion Markup Language (SAML) 2.0 Technical Overview*. Organization for the Advancement of Structured Information Standards (OASIS), junho de 2005d.

OMG. *The Common Object Request Broker Architecture v3.0.2*. Object Management Group (OMG), dezembro de 2002.

OpenGroup. *DCE 1.1: Remote Procedure Call*. Open Group Technical Standard, AE Specification C309, agosto de 1997.

Papazoglou, M. P. "Service-oriented computing: Concepts, characteristics and directions". Em *4th International Conference on Web Information Systems Engineering (WISE'03)*, 2003.

Patil, V. e Shyamasundar, R. "Trust Management for e-Transactions". *Sadhana*, 30(2 and 3):141–158, abril de 2005.

Peersim. "Peersim P2P Simulator", 2004. `http://peersim.sourceforge.net`.

Penning, H. P. "Analysis of the strong set in the PGP web of trust", 2006. `http://www.cs.uu.nl/people/henkp/henkp/pgp/pathfinder/plot/`.

Rannenberg, K. "Multilateral security a concept and examples for balanced security". Em *Workshop on New security paradigms (NSPW'00)*, pp. 151–162, New York, NY, USA, 2000. ACM Press. ISBN 1-58113-260-3.

Rivest, R. L. e Lampson, B. "SDSI – A Simple Distributed Security Infrastructure". Presented at CRYPTO'96 Rumpsession, 1996.

Rowstron, A. e Druschel, P. "Pastry: scalable, decentraized object location and routing for large-scale peer-to-peer systems". Em *18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, novembro de 2001.

RSA. *PCKS#1 v2.1: RSA Cryptography Standard*. RSA Laboratories, junho de 2002. `ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-1/pkcs-1v2-1.pdf`.

RSS. *Really Simple Syndication*. RSS Advisory Board, janeiro de 2005. `http://www.rssboard.org/rss-specification`.

Russell, D. e Gangeni, G. *Computer Security Basics*. O'Reilly Associates Inc., 1991.

Sabater, J. e Sierra, C. "Regret: A reputation model for gregarious societies". *4th Workshop on Deception, Fraud and Trust in Agent Societies*, pp. 61–69, 2001.

Sabater, J. e Sierra, C. "Review on Computational Trust and Reputation Models". *Artificial Intelligence Review*, 24(1):33–60, 2005.

Santin, A. O. *Teias de federações: uma abordagem baseada em cadeias de confiança para autenticação, autorização e navegação em sistemas de larga escala*. Tese de doutorado, Universidade Federal de Santa Catarina, 2004.

Santin, A. O., da Silva Fraga, J., Siqueira, F., e de Mello, E. R. "Federation Web: A Scheme to Compound Authorization Chains on Large-Scale Distributed Systems". Em *22nd Symposium on Reliable Distributed Systems (SRDS'03)*, Florença - Itália, 2003.

Seigneur, J.-M., Farrell, S., e Jensen, C. D. "Secure Ubiquitous Computing based on Entity Recognition". Em *Workshop on Security in Ubiquitous Computing (UBICOMP'02)*, setembro de 2002.

Shibboleth. *Shibboleth Architecture*, junho de 2005. `http://shibboleth.internet2.edu/docs/draft-mace-shibboleth-tech-overview-latest.pdf`.

Singhal, A., Winograd, T., e Scarfone, K. "Guide to Secure Web Services". *NIST Special Publication*, (800-95), agosto de 2007.

Skogsrud, H., Benatallah, B., e Casati, F. "Modelo-Driven Trust Negotiation for Web Services". Em *IEEE Internet Computing*, pp. 45–52. IEEE Computer Society, dezembro de 2003.

Smith, M. *Definition of the inetOrgPerson LDAP Object Class*. IETF RFC 2798, abril de 2000.

Spantzel, A. B., Squicciarini, A. C., e Bertino, E. "Integrating federated identity management and trust negotiation". Relatório Técnico 2005-46, CERIAS – Purdue University, 2005.

Stoica, I., Morris, R., Liben-Nowell, D., Karger, D. R., Kaashoek, M. F., Dabek, F., e Balakrishnan, H. "Chord: a scalable peer-to-peer lookup protocol for internet applications". *IEEE/ACM Transaction on Networking*, 11(1):17–32, 2003.

Sun. "Java Remote Method Invocation Specification". Revision 1.8 Java 2 SDK, 2002.

Sutton, R. S. e Barto, A. G. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

Tangmunarunkit, H., Govindan, R., Jamin, S., Shenker, S., e Willinger, W. "Network topology generators: degree-based vs. structural". Em *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication (SIGCOMM'02)*, pp. 147–159. ACM, 2002.

Teacy, W. T., Patel, J., Jennings, N. R., e Luck, M. "TRAVOS: Trust and Reputation in the Context of Inaccurate Information Sources". *Autonomous Agents and Multi-Agent Systems*, 12(2):183–198, 2006. ISSN 1387-2532.

Vogels, W. "Web Services are Not Distributed Objects". *Internet Computing*, 7(6):59–66, novembro de 2003.

W3C. *Web Services Description Language 1.1*. W3C Working Group, março de 2001.

W3C. *The Platform for Privacy Preferences 1.0 (P3P1) Specification*. W3C Recommendation, abril de 2002. `http://www.w3c.org/TR/P3P`.

W3C. *SOAP 1.2 – W3C Recommendation*. W3C, junho de 2003. http://www.w3.org/TR/soap12.

W3C. *Web Services Architecture*. W3C Working Group, fevereiro de 2004. http://www.w3.org/TR/2004/NOTE-ws-arch-20040211.

Wahl, M. *A Summary of the X.500(96) User Schema for use with LDAPv3*. IETF RFC 2256, dezembro de 1997.

Wang, Y. e Vassileva, J. "Bayesian Network Trust Model in Peer-to-Peer Networks". *Workshop on Deception, Fraud and Trust in Agent Societies*, 7, 2003.

Wangham, M. S., Boger, D., de Mello, E. R., e da Silva Fraga, J. "Implementação de um Serviço XKMS para o SPKI/SDSI com suporte as Federações SPKI". Em *Salão de Ferramentas do SBRC'07*, Belém, PA, 2007.

Weerawarana, S., Curbera, F., Leymann, F., Storey, T., e Ferguson, D. F. *Web Services Plataform Architecture*. Prentice Hall, março de 2005.

Whitby, A., Jøsang, A., e Indulska, J. "Filtering out unfair ratings in bayesian reputation systems". *The Icfain Journal of Management Research*, 4(2):48–64, 2005.

Winslett, M., Yu, T., Seamons, K. E., Hess, A., Jacobson, J., Jarvis, R., Smith, B., e Yu, L. "Negotiating Trust on the Web". Em *IEEE Internet Computing*, volume 6, pp. 30–37. IEEE Computer Society, dezembro de 2002.

WS-Federation. *Web Services Federation Language*, julho de 2003. http://msdn.microsoft.com/ws/2003/07/ws-federation.

WS-Policy. *Web Services Policy Framework*, setembro de 2004. http://msdn.microsoft.com/ws/2004/09/policy/.

WS-Trust. *Web Services Trust Language (WS-Trust)*, fevereiro de 2005. http://msdn.microsoft.com/library/en-us/dnglobspec/html/WS-Trust.asp.

Wu, T. "The Secure Remote Password Protocol". Em *Internet Society Network and Distributed System Security Symposium*, pp. 97–111, 1998.

Yang, B. e Garcia-Molina, H. "Improving Search in Peer-to-Peer Networks". Em *22nd International Conference on Distributed Computing Systems (ICDCS'02)*, pp. 5–14, 2002.

Yavatkar, R., Pendarakis, D., e Guerin, R. *A Framework for Policy-based Admission Control*. IETF RFC 2753, janeiro de 2000.

Zhang, H. e van Moorsel, A. "Evaluation of P2P Algorithms for Probabilistic Trust Inference in a Web of Trust". Relatório Técnico CS-TR-1113, Newcastle University, julho de 2008.

Zhuang, Z., Liu, Y., Xiao, L., e Ni, L. M. "Hybrid Periodical Flooding in Unstructured Peer-to-Peer Networks". Em *ICPP*, pp. 171–178. IEEE Computer Society, 2003. ISBN 0-7695-2017-0.

Zimmerman, P. *PGP User's Guide*. Massachusetts Institute of Technology, maio de 1994.